Tracing the evolution of teachers’ mathematical knowledge and pedagogy through programming

Learning from Scratch

Piers Saunders

Supervisors:
Professor Dame Celia Hoyles, Dr Manolis Mavrikis,
Professor Richard Noss, Professor David Pratt.

Culture, Communication and Media
University College London, Institute of Education.

This thesis is submitted for the degree of

Doctor of Philosophy
I would like to dedicate this thesis to my parents and my best friend Nancy who has supported me all the way, but more importantly has checked in on me every day during the COVID-19 pandemic.
Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains approximately 100,000 words excluding table of contents, references and appendices.

Piers Saunders
2022
Abstract

This thesis is based on research to explore the role of primary school teachers’ mathematical and pedagogical knowledge in their engagement with computer-based microworlds that formed part of ScratchMaths (SM). SM is a two-year mathematics and computing curriculum designed for pupils aged nine to eleven years old. The aims of the research were to trace the evolution of teachers’ mathematical knowledge, as they taught SM microworlds designed for exploration and reasoning about place value, variable and angle through computer programming.

The study adopted a multiple-case study approach with augmenting teacher episodes situated in the English primary school setting. The thirteen Year 6 teachers of the study were selected from national participants of the second year of the two-year SM intervention. Data collection involved video-recorded classroom observations, audio-recorded post-lesson semi-structured teacher interviews, and ‘think aloud’ while engaging with computer-based tasks. The conceptual framework for the thesis incorporated the Mathematical Pedagogical Technology Knowledge (MPTK) framework and the Instrumental Orchestration model. The findings reveal the knowledge required to teach at the intersection of programming and mathematics, and crucially, how the ideas mediate and are mediated by engagement with the SM curriculum. The findings also illustrate how teaching mathematics through computer programming requires the teacher to bridge between the computing and the mathematics domains and how some teachers managed to do this while creating new connections within and between the knowledge domains.

The study contributes to the literature of teachers’ mathematical knowledge of place value, variable, and angle as well as teachers’ ability to (re-) express mathematics through computer programming. The thesis makes an original contribution to the literature with the specification of a theoretical model for analysing teachers’ knowledge for teaching mathematics through programming in the primary setting.
Impact statement

The aim of this thesis is to explore the role of primary school teachers’ mathematical and pedagogical knowledge in their engagement with computer-based microworlds that formed part of ScratchMaths (SM). This research presents windows on teachers’ mathematical pedagogical technology knowledge (MPTK) as they engage with and teach the Scratch microworlds designed for exploration and reasoning about place value, variable and angle through computer programming. The research provides a detailed analysis of the evolution of the teachers’ mathematical knowledge, their pedagogical practices and, crucially, how the ideas mediate and are mediated by engagement with the SM curriculum. Thus, the research shows how teachers’ mathematical and pedagogical knowledge evolve together as they teach mathematics through programming.

The SM intervention was evaluated in 2018 by the Education Endowment Foundation (EEF) who found that engagement with SM had a positive impact on computational thinking but had no impact on mathematical attainment as measured by pupils’ performance in the Key Stage 2 test. This was disappointing, altogether not surprising given the low fidelity of curriculum implementation of SM within Year 6 (Noss et al., 2020). My research findings have an important role in telling the complete story of SM, by its focus on teachers, the criticality of their role and the impact of practice on their knowledge development.

The findings of the research exemplify the knowledge required to teach at the intersection of programming and mathematics, which goes beyond knowledge in each domain. Teaching mathematics through computer programming requires the teacher to bridge between the computing and the mathematics domains. The findings illustrate how some teachers managed to do this while creating new connections within and between the knowledge domains. Others showed limitations in their mathematical knowledge of place value, variable, and angle as well as in their ability to (re-) express mathematics through programming. Both aspects are of interest to the wider mathematics education community.

All of the findings highlight the essential role of professional development (PD) to support teachers’ evolving knowledge and pedagogy to teach mathematics through programming. Any PD must go beyond just learning how to program as highlighted by the teaching episodes where limited PD seemed to result in unproductive and possibly harmful learning outcomes for the pupils. This research therefore has the potential to influence the design of curricula materials and corresponding professional development to meet the aims of a National Computing Curriculum in using programming in mathematics but also other subject domains.

I have shared some of my results with researchers and educational practitioners to maximise the impact for knowledge exchange. As an example, I am currently working on an ESRC-MOST UK-Taiwan Networking project. The 18-month collaboration (started June 2021) between UCL IOE and National Taiwan Normal University aims to explore the use of selected...
SM materials to teach English-language STEM education in Taiwanese schools. The project will form part of a proposal for a wider ESRC-MOST project in 2021-2022.

Some of the research has been disseminated at both academic and professional conferences nationally and discussed in the following publications:

Acknowledgements

I would like to acknowledge the academic support, the encouragement, the guidance and the motivation from my supervisors during the past eight years of doctoral study: Professor David Pratt, Professor Richard Noss, Professor Dame Celia Hoyles and Dr Manolis Mavrikis.

Celia and Richard have been instrumental in my development through academic life. I first encountered their research with Logo at the Institute of Education in 1994 where I learned to become a teacher of mathematics. Their work influenced so much of my fascination with using technology in the classroom as well as my love of mathematics. However, it was not until sixteen years later at the start of my master’s programme that I was able to meet them in person. It was an honour to be able to work with Celia and Richard during the ScratchMaths project and to have their supervision during my doctoral study. I would not have reached the end without their passion for mathematics and technology, their faith in my research, their words of encouragement and their constant presence in my life.

Thank you, Celia and Richard.
Contents

Contents ........................................................................................................................................... viii
List of Figures ................................................................................................................................... xiii
List of Tables ...................................................................................................................................... xix
Nomenclature and acronyms ........................................................................................................... 1
Chapter 1 Introduction ..................................................................................................................... 3
  1.1 Rationale for the study ................................................................................................................ 3
  1.2 Personal motivations for the study ............................................................................................ 5
  1.3 Introducing the ScratchMaths project ....................................................................................... 7
  1.4 Structure of the thesis ............................................................................................................... 10
Chapter 2 Literature review ............................................................................................................ 13
  2.1 Introduction ............................................................................................................................... 13
  2.2 Mathematical foci of the study: mathematical thinking and reasoning, place value and variable ......................................................................................................................... 14
    2.2.1 Research on the teaching and learning of mathematical thinking and reasoning 14
    2.2.2 Research on the teaching and learning of place value in mathematics ..................... 16
    2.2.3 Research on the teaching and learning of algebraic variable in mathematics .......... 23
  2.3 Teaching and learning mathematics through programming .................................................. 26
    2.3.1 Introducing Logo and representing mathematics through programming .............. 26
    2.3.2 Research on the teaching and learning of angle ............................................................... 30
    2.3.3 Research on teaching and learning of variable in computer programming .............. 32
    2.3.4 Introducing Scratch, a blocks-based Logo programming environment .................. 34
    2.3.5 Computational thinking ..................................................................................................... 36
    2.3.6 The intersection of computational thinking and mathematical thinking ................ 39
    2.3.7 The ScratchMaths 5E pedagogic framework ................................................................. 41
  2.4 Towards a conceptual framework for analysing teachers’ knowledge for teaching mathematics through programming in the primary setting ......................................................... 45
    2.4.1 Mathematical knowledge for teaching (MKT) framework ............................................ 46
    2.4.2 Teacher knowledge for teaching mathematics with technology frameworks .......... 47
    2.4.3 Mathematical Pedagogical Technology Knowledge (MPTK) framework for teaching mathematics through programming in the primary setting ........................................ 52
  2.5 Research questions ................................................................................................................... 54
Chapter 3 Research methodology and methods ............................................................................. 59
  3.1 Introduction ............................................................................................................................... 59
3.2 Research paradigm and design of the study ........................................... 59
3.3 Selecting participants for the study ....................................................... 63
  3.3.1 The case study teachers ................................................................. 65
3.4 Methods of data collection ................................................................... 70
  3.4.1 Classroom observation ................................................................. 70
  3.4.2 Teacher interviews ................................................................. 73
  3.4.3 Labelling and managing data ...................................................... 82
  3.4.4 Challenges during data collection .................................................. 83
3.5 Operationalising my framework for teaching mathematics through programming in the primary setting .................................................. 85
3.6 Data analysis ..................................................................................... 89
  3.6.1 Developing an analytical approach for data analysis ......................... 89
  3.6.2 Piloting the analytical approach .................................................... 92
  3.6.3 Issues of validity ........................................................................ 93
  3.6.4 Teaching for robust understanding in the context of teaching mathematics through programming in the primary setting ........................................... 95
3.7 Ethical considerations ...................................................................... 98
3.8 Summary ......................................................................................... 100

Chapter 4 The ScratchMaths curriculum and microworlds ................................ 101
4.1 Introduction ..................................................................................... 101
4.2 ScratchMaths curriculum .................................................................. 101
  4.2.1 Module 4 – Building with numbers - Place value microworld .......... 102
  4.2.2 Module 5 – Exploring mathematical relationships - Fireworks microworld .... 108
  4.2.3 Teacher support materials ............................................................ 118
4.3 Case study structure and emergent themes ......................................... 121

Chapter 5 Case study 1 – Mary (MW) .................................................. 125
5.1 Introduction ..................................................................................... 125
5.2 Window on mathematical knowledge for teaching ............................... 125
  5.2.1 Window on place value ............................................................ 125
  5.2.2 Window on variable ................................................................ 128
  5.2.3 Window on angle ...................................................................... 133
5.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum ............................................. 135
  5.3.1 Thinking with your maths head on ............................................. 135
  5.3.2 Thinking like a sprite ................................................................. 138
5.4 Teacher pedagogic practices ............................................................... 145
  5.4.1 Whole-class orchestrations ......................................................... 145
  5.4.2 Guide-and-Explain orchestration ............................................... 150
5.4.3 Link-multi-modal-representation orchestration ........................................ 151
5.4.4 The TRU framework .................................................................................... 154
5.5 Summary ........................................................................................................ 158
Chapter 6 Case study 2 – Rina (RE) ................................................................. 163
6.1 Introduction .................................................................................................... 163
6.2 Window on mathematical knowledge for teaching ........................................ 164
   6.2.1 Window on place value ............................................................................ 164
   6.2.2 Window on variable ............................................................................... 166
   6.2.3 Window on angle .................................................................................... 171
6.3 How mathematical ideas mediate and are mediated by engagement with the
   ScratchMaths curriculum .............................................................................. 174
   6.3.1 Thinking with your maths head on ......................................................... 175
   6.3.2 Thinking like a sprite ............................................................................. 179
6.4 Teacher pedagogic practices ....................................................................... 189
   6.4.1 Whole-class orchestrations .................................................................... 189
   6.4.2 Spot-and-Show orchestration ................................................................. 194
   6.4.3 Discuss-the-screen orchestration ............................................................ 196
   6.4.4 The TRU framework ............................................................................ 199
6.5 Summary ........................................................................................................ 205
Chapter 7 Case study 3 – Sally (SE) ................................................................. 209
7.1 Introduction .................................................................................................... 209
7.2 Window on mathematical knowledge for teaching ........................................ 209
   7.2.1 Window on variable ............................................................................... 209
   7.2.2 Window on angle .................................................................................... 219
7.3 How mathematical ideas mediate and are mediated by engagement with the
   ScratchMaths curriculum .............................................................................. 223
   7.3.1 Thinking with your maths head on ......................................................... 223
   7.3.2 Thinking like a sprite ............................................................................. 227
7.4 Teacher pedagogic practices ....................................................................... 232
   7.4.1 Whole-class orchestrations .................................................................... 232
   7.4.2 Link-multi-modal-representation orchestration ...................................... 235
   7.4.3 The TRU framework ............................................................................ 240
7.5 Summary ........................................................................................................ 244
Chapter 8 Teacher episodes ........................................................................... 249
8.1 Introduction .................................................................................................... 249
8.2 Window on mathematical knowledge for teaching ........................................ 249
   8.2.1 Window on place value ............................................................................ 249
8.2.2 Window on variable ........................................................................................................... 253
8.2.3 Window on angle ............................................................................................................. 267
8.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum ........................................................................................................... 269
8.3.1 Thinking with your maths head on .................................................................................. 270
8.3.2 Thinking like a sprite ...................................................................................................... 274
8.4 Teacher pedagogic practices .............................................................................................. 279
8.4.1 Whole class orchestrations ............................................................................................ 280
8.4.2 The TRU framework ..................................................................................................... 284
8.5 Summary ............................................................................................................................ 291
Chapter 9 Findings and discussion .......................................................................................... 297
9.1 Introduction ......................................................................................................................... 297
9.2 An overview of the research study ...................................................................................... 297
9.3 The key findings .................................................................................................................. 298
9.4 Findings related to teachers’ evolving knowledge (RQ1) .................................................... 301
9.4.1 Teaching and building place value models in ScratchMaths supported teachers to examine their mathematical knowledge for teaching place value and to develop new knowledge connections ............................................................................ 301
9.4.2 Teaching using the ScratchMaths fireworks microworld supported teachers to develop their knowledge of variable within and between contexts ................................................................. 304
9.4.3 Teaching using the ScratchMaths fireworks microworld provided teachers with opportunities to confront and reorganise their mathematical knowledge of angle .............................................. 308
9.5 Discussion and summary on the findings related to teachers evolving knowledge. 309
9.6 Findings related to teachers’ evolving pedagogy (RQ2) .................................................... 311
9.6.1 Teachers developed similes such as thinking like a sprite or thinking with their Scratch/Maths head on as a way to effectively bridge between mathematical and computational ideas. .................................................................................. 311
9.6.2 Teachers used multimodal representations to effectively bridge between computing and mathematics using multiple non-computer-based representations. ..... 315
9.6.3 Teachers used a variety of whole-class orchestrations. As their MPTK developed they frequently used orchestrations which involved using Scratch live. ................................. 317
9.6.4 Teachers created well-managed and structured classrooms to provide opportunities for children to effectively learn mathematics through programming. ..... 321
9.6.5 Teachers who frequently used each and all of the 5E’s from the pedagogic framework, provided opportunities for effective learning of mathematics through programming .......................................................................................................................... 323
9.6.6 Professional development is essential to support teachers’ development of MPTK. 324
9.7 Discussion and summary of the findings related to teachers evolving pedagogy. 325
Chapter 10 Conclusion .............................................................................................................. 329
List of Figures

Figure 2.1 A model of the framework for PTK (Thomas and Palmer, 2014, p. 76)...........49
Figure 2.2 Mathematical Pedagogical Technology Knowledge (MPTK) framework for teaching mathematics through programming in the primary setting ..................................................54
Figure 3.1 The five dimensions of robust classrooms - the teaching for robust understanding (TRU) framework (Schoenfeld, 2018, p. 493).................................................................96
Figure 4.1 Ones sprite, costumes and next costume block to simulate counting by 1 ........103
Figure 4.2 Script for Tens sprite to listen for the add ten message..........................104
Figure 4.3 Script for Ones sprite to broadcast a message using if then structure and costume # reporter block ...............................................................................................105
Figure 4.4 Script to check if costume # = 1 to represent the place value carry/exchange mechanism ..................................................................................................................106
Figure 4.5 Script to check if costume # = 10 to represent the place value carry/exchange mechanism ..................................................................................................................107
Figure 4.6 ScratchMaths 4-digit place value model project ......................................108
Figure 4.7 Fireworks microworld beetle sprite viewed from above (analogous to Papert's turtle). .................................................................109
Figure 4.8 Scratch script to draw a square, a pentagon and a hexagon .....................109
Figure 4.9 General regular polygon script using answer block to represent number of sides .................................................................110
Figure 4.10 Scratch interface showing the beetle asking a question, (storing the value), and then reporting the value of answer using a say block to join together the text hello and answer. .................................................................111
Figure 4.11 ScratchMaths activity 5.1.3 - Naming Values – PowerPoint slide.............114
Figure 4.12 Script to draw a regular polygon which asks for two user inputs and stores them in two variables: number of sides and side length ........................................................................115
Figure 4.13 ScratchMaths polygon fireworks in the night sky Scratch stage output ......115
Figure 4.14 SM Activity 5.2.1: Sequence of Squares................................................116
Figure 4.15 Scratch script to produce a sequence of squares ....................................117
Figure 4.16 ScratchMaths exemplar teacher guide for activity instructions from Module 5 – Exploring Mathematical Relationships - Investigation 1 – Activity 5.1.1: Ask and Answer (UCL ScratchMaths, 2017) .................................................................119
Figure 4.17 ScratchMaths exemplar teacher guide for additional support from Module 5 – Exploring Mathematical Relationships - Investigation 1 – Activity 5.1.1: Ask and Answer (UCL ScratchMaths, 2017) ................................................................. 120

Figure 5.1 Polygon definition block to draw a polygon using variables for number of sides and side length ........................................................................................................ 129

Figure 5.2 Script to draw a regular polygon which asks for two user inputs and stores them in two variables: number of sides and side length .............................................. 130

Figure 5.3 Mary's script for the task “Build a script which asks for the number of sides, and then draws a polygon of that number of sides” ...................................................................... 131

Figure 5.4 Mary's response to creating a script which asks for the side length, numbers of sides, sets side length and number of sides variables and then draws a regular polygon and displays its perimeter ............................................................................................................. 132

Figure 5.5 Child's screen as they tried to program the beetle to draw an equilateral triangle ........................................................................................................................................ 133

Figure 5.6 Mary's white board explaining the exterior angle of a triangle. She had marked 120 degree external angle and 60 degree interior angle. The right hand side shows the output of move 60 three times .................................................................................................................................. 134

Figure 5.7 SM Activity 4.1.2: unplugged flip book Flip Flip Nudge Nudge game .................. 136

Figure 5.8 SM Activity 4.1.1: modify the script so that when the ones sprite is clicked it changes it value in different ways .............................................................................................................. 139

Figure 5.9 Script to broadcast a message and mimic the nudge behaviour ................................ 140

Figure 5.10 SM Activity 5.1.1: introducing ask and answer blocks ........................................... 141

Figure 5.11 Scratch display of child's solution who had hard coded their name, rather than using the answer block ........................................................................................................ 142

Figure 5.12 SM Activity 5.1.1 to draw a regular polygon of any pen size and a random colour ................................................................................................................................. 144

Figure 5.13 Child’s script to draw an equilateral triangle using a turn of 60 degrees .......... 144

Figure 5.14 SM Activity 5.1.3: consider if a script using two ask blocks and one answer block will draw a regular polygon of any number of sides and side length ........................................ 149

Figure 5.15 Mary pointing to the tens sprite on the interactive whiteboard. The children are building the countdown timer alongside Mary, she guides and explains each step. ........... 150

Figure 5.16 Children using place value flip books to think like a sprite as they engage with the countdown timer activity ........................................................................................................ 152

Figure 5.17 Chart to show weighted averages for each dimension of Mary's eight observed lessons using the adapted TRU framework for Maths through programming ....................... 155

Figure 5.18 Script to draw 5 squares of increasing size as the variable increases................. 156

Figure 6.1 Rina's response to a gap task - Draw a scene containing houses ........................ 163

Figure 6.2 SM Activity 5.1.3: Discuss if the given solution will work, a script which asks two questions and uses the answer block ................................................................. 168

Figure 6.3 SM Activity 5.1.3: Instructions to create variables for side length and number of sides and explore their use ............................................................................................... 169
Figure 6.4 Rina's project containing two questions asked and two set variables and a simple polygon script to be edited
Figure 6.5 Simple script to draw a polygon using default blocks dragged from the Scratch palette
Figure 6.6 SM Activity 5.2.2: Altering rectangles
Figure 6.7 Rina's general rectangle script and output using a Height and a Base variable
Figure 6.8 Rina’s screen to show the result of a rectangle drawn of height 360 which goes out of screen bounds
Figure 6.9 SM Activity 5.2.2: unpluged activity to match rectangle patterns with their corresponding scripts
Figure 6.10 Rina's screen displaying the broken stopwatch and the code inside the 10 seconds sprite
Figure 6.11 Defined square block using side length variable to control move
Figure 6.12 Scratch exemplar scripts to draw towers of squares with side lengths of 5, 10 and 15
Figure 6.13 Rina demonstrating the starting and finishing place for the beetle on the flip chart
Figure 6.14 Rina's flipchart showing a square, four labelled sides marked with length sl and a move block with sl inside it
Figure 6.15 Rina’s white board to show worded problems and corresponding mathematical expression
Figure 6.16 SM Activity 5.2.2 – Altering Rectangles
Figure 6.17 Rina acting out playing beetle to draw the growing rectangle pattern using a variable for height
Figure 6.18 Script for one seconds sprite in the stopwatch activity
Figure 6.19 Rina live coding using the computer sitting beside the IWB in the Discuss-the-screen orchestration. Figure 6.20 below shows the script she was building
Figure 6.20 Rina’s script to draw a polygon for any number of sides containing a bug
Figure 6.21 Updated script to include the pick random for the go to x: y
Figure 6.22 Chart to show weighted averages for each dimension of Rina's observed lessons using the adapted TRU framework for Maths through programming
Figure 6.23 SM Activity 5.2.1: Sequence of Squares
Figure 6.24 Rina’s white board to show the output of the script as the beetle moves to draw square patterns
Figure 7.1 Completed Polygon Fireworks scratch project
Figure 7.2 Revised polygon firework script using one answer block to vary the number of side of the polygon and reduce complexity
Figure 7.3 Sally’s creates slide which asks the children to consider with which polygon is most challenging to draw in Scratch
Figure 7.4 Sally marking the corner of of a polygon with a circle to represent the angle turned
Figure 7.5 SM graphic to explain exterior and interior angle as drawn by the beetle taken from the ScratchMaths teacher guide........................................................................................................222
Figure 7.6 SM Activity 5.2.1 - Use the square block and set side length to draw patterns of squares with different sizes........................................................................................................224
Figure 7.7 Scratch interface to show script for the Altering Polygons activity which uses set side length to vary the side length of a defined square block..................................................................................226
Figure 7.8 SM Activity 5.2.1: Sequence of Squares........................................................................................................228
Figure 7.9 Scratch display showing set, square, and change blocks joined together. ........228
Figure 7.10 Sally using a cut out beetle to represent the beetle's movement as it draws a square .................................................................................................................................................229
Figure 7.11 Defined square block using side length variable......................................................231
Figure 7.12 SM Activity 5.1.3: consider if a script using two ask blocks and one answer block will draw a regular polygon of any number of sides and side length ........................................236
Figure 7.13 Scratch project to show isolated scripts which set side length variable to different values and then use it to draw a square........................................................................................................237
Figure 7.14 Sally holding a four by four multilink square to link the representation of a square drawn on the screen, the Scratch script and the drawing on the white board........................................238
Figure 7.15 Sally showing the cut out beetle moving the side length as represented on the white board .................................................................................................................................................239
Figure 7.16 Chart to show weighted averages for each dimension of Sally's observed lessons using the adapted TRU framework for Maths through programming ..........................................240
Figure 8.1 SM Activity 5.1.1 - Introducing the Ask and Answer blocks ........................................253
Figure 8.2 Example of child's work which does not use the answer block inside the say block. .................................................................................................................................................254
Figure 8.3 Example of Betty's script using if then and answer ......................................................254
Figure 8.4 Example of Betty's script using if, then, else and answer .............................................255
Figure 8.5 Matt’s initial script to draw a regular polygon which uses an ask block and an expression using answer to control the turn ..............................................................................................................................................263
Figure 8.6 Matt’s developing polygon script which asks a second question question for how many sides inside the repeat ..........................................................................................................................264
Figure 8.7 Matt’s script to represent the general structure of a regular polygon, a repeat, a move and a turn ..................................................................................................................................................264
Figure 8.8 Matt exploring where to place the ask block as he confronts the limitation of not being able to store two answers simultaneously ......................................................................................265
Figure 8.9 Matt’s final script to draw a regular polygon and display its perimeter using one variable for the number of sides and a fixed side length of 20 .................................................................................266
Figure 8.10 Child's ScratchMaths polygon fireworks and towers. ................................................268
Figure 8.11 Four-digit place value model used by Katherine edited to contain a bug, e.g., if costume # = 9 ..................................................................................................................................................270
Figure 8.12 Scratch stage showing costume # reporter block reporting a value of 10 for the digit costume (0) worn by the ones sprite..............................................................................................................272
Figure 8.13 Activity 4.2.1 from SM - Build a stopwatch investigation.................................273
Figure 8.14 Script which asks four questions to demonstrate how each successive ask block overwrites the previous value of answer stored.................................................................275
Figure 8.15 Post-it Note as held by the children (L to R) Adam: How many sides; Brianna: What side length; C: Chelsea: What pen size; Dave: How many polygons. .........................276
Figure 8.16 Group A scores across 5 dimensions of TRU framework for one observed lesson. ........................................................................................................................................286
Figure 8.17 Group B scores across 5 dimensions of TRU framework for one observed lesson. ........................................................................................................................................286
Figure 9.1 Screen output when drawing near the limits of the Scratch stage.........................314
Figure 10.1 Mathematical Pedagogical Technology Knowledge framework for teaching mathematics through programming in the primary setting..............................................338
Figure 10.2 An image of the Scratch stage with a sprite on it...............................................I-27
List of Tables

Table 2.1 Mapping of i) Brennan and Resnick's CT framework, ii) Cuoco, Goldenberg and Mark’s Mathematical Habits of mind and iii) ScratchMaths 5Es pedagogical framework......44

Table 3.1 Teachers, schools and profiles for Group A and Group B of the case study sketches ..................................................................................................................................................69

Table 3.2 Place value mathematical content knowledge items used in teacher interviews.....76

Table 3.3 Variable task and reflection questions used in teacher interviews ..........................81

Table 3.4 Table of codes by local area, school and teacher........................................................82

Table 3.5 Data source codes used in thesis..................................................................................83

Table 3.6 Mapping data collection methods against the MPTK framework ..............................86

Table 3.7 A model for operationalising the MPTK framework to examine teachers’ classroom practice using the ScratchMaths curriculum..................................................................................88

Table 3.8 Summary of code frequency after pilot coding two lesson observations ...............93

Table 4.1 ScratchMaths overview of modules containing mathematics and computing concepts ............................................................102

Table 4.2 Costume # for the ordered list of costumes when costume 0 is placed first or placed last........................................................................................................................................106

Table 4.3 Series of scripts which show the progression of distancing using and operating on variable from the initial ask block which sets the variable value .................................................................113

Table 4.4 Reasoning steps to produce a sequence of squares......................................................118

Table 4.5 Classification of codes and properties of codes to exemplify the two themes related to teachers use of similes which emerged from analysis of lesson observations...........................122

Table 4.6 Mapping of research questions to aspect of the MPTK framework and case study section ........................................................................................................................................123

Table 5.1 Characterization and frequency of whole-class orchestration types as observed in Mary's lessons .........................................................................................................................................147

Table 5.2 Weighted averages for each dimension of Mary's eight observed lessons using the adapted TRU framework for Maths through programming.................................................................155

Table 5.3 Table to show summary of findings and location within the chapter for Mary's case study ........................................................................................................................................160

Table 6.1 Rina explaining her current understanding of the concept of variable....................167

Table 6.2 Whole-class activity types, their characterisation and frequency observed across the six observed lessons ........................................................................................................................................191
Table 6.3 Weighted averages for each dimension of Rina’s observed lessons using the adapted TRU framework for Maths through programming ......................................................... 200
Table 6.4 Table to show summary of findings and location within the chapter for Rina's case study .................................................................................................................................................. 207
Table 7.1 Extracts of Sally thinking out loud as she worked on the interview based Scratch task .................................................................................................................................................. 216
Table 7.2 Whole-class activity types, their characterisation and frequency observed across the four observed lessons ........................................................................................................ 233
Table 7.3 Weighted averages for each dimension of Sally’s observed fireworks microworld lessons using the adapted TRU framework for Maths through programming ...................................... 240
Table 7.4 Table to show summary of findings and location within the chapter for Sally’s case study .................................................................................................................................................. 247
Table 8.1 Sequence of Nancy thinking out loud as she worked on the interview based Scratch task .................................................................................................................................................. 259
Table 8.2 Cumulative time for each whole-class orchestration in one lesson .................................................................................................................................................. 280
Table 8.3 Scores for each dimension of an observed microworld lessons using the adapted TRU framework for Maths through programming .................................................................................. 285
Table 8.4 Table to show summary of findings and location within the chapter for the Stetches case study .................................................................................................................................................. 294
Table 9.1 Table to show frequency of each orchestration for all thirteen teachers from the study. Group A (Betty) means Betty from this group only .................................................................................................................. 318
Table 10.1 Adapted Teaching for Robust Understanding (TRU) summary (Schoenfeld, 2014a; Schoenfeld, 2018) scoring rubric used for classroom analysis of Teaching Mathematics through programming .................................................................................................................................................. 341
Table 10.2 ScratchMaths teachers observed across each Microworld. Each observed lesson length was between 60 and 100 minutes .................................................................................................................................................. H-26
Nomenclature and acronyms

Scratch is a free programming language developed by MIT that makes it easy to create interactive stories, animations, games, music, and art, and share your creations on the web. Scratch can run from within a modern web browser or downloaded as an app.

ScratchMaths is a two-year mathematics and computing curriculum designed for pupils aged nine to eleven years old.

**Bold** text indicates the use of a Scratch block e.g., *move, turn, broadcast, variable, ask, answer, when I receive, repeat.*

MPTK – Mathematical Pedagogical Technology Knowledge

SM - ScratchMaths

PD - Professional development
Chapter 1    Introduction

This thesis brings together the fields of teacher knowledge for teaching mathematics, and teacher knowledge for teaching computer programming to explore teaching mathematics through computer programming. Using classroom observations, interviews, and task-based interviews, this thesis traces the evolution of teachers’ knowledge and how it is involved in interacting with computer programming to express mathematical ideas. This contributes towards a broader aim to develop a deeper understanding of the knowledge at the intersection of computer programming and mathematics as teachers engage with using computer programming to teach a specially designed curriculum called ScratchMaths.

The introductory chapter provides the contextual background to situate the study, my personal motivations for undertaking the research, an overview of the ScratchMaths project and the structure of the thesis.

1.1    Rationale for the study

Klaus Schwab (2017) defined the Fourth Industrial revolution to represent a fundamental change in the way we live, work and relate to one another. He argued that the new period of human development is enabled by extraordinary technological advances commensurate with the previous industrial revolutions. Schwab describes the “unlimited possibilities of having billions of people connected by mobile devices” and the “emerging technology breakthroughs, covering wide-ranging fields such as artificial intelligence (AI), robotics, the internet of things (IoT)” (Schwab, 2017, p. 7). He explains that the real opportunity is to look beyond technology and find ways to positively impact the greatest number of people. At the official conclusion of a 2013 summit in Brussels, a member of parliament stated that the European Union was faced with 300,000 unfilled vacancies in the information and communications technology (ICT)
sector and it was noted that if this trend is not checked, there could be as many as 900,000 unfilled vacancies by 2015 (Parliament. House of Lords, 2013). As a result of this concern a new Computer Science curriculum was proposed for all students in primary and secondary education. The new curriculum would shift the focus of the information and communications technology (ICT) curriculum to a new Computer Science curriculum which would have an effect of preparing students for the demands of further study. The ICT curriculum was frequently acknowledged as unambitious, demotivating, and dull as it focused on using ICT applications such as Word and Excel. However, the new computing curriculum would provide opportunities for eleven year olds to write simple 2D computer animations, or sixteen year olds to write apps for smart phones (Department for Education, 2012). Central to the new computer science curriculum would be the role of computer programming.

From September 2014 all primary (and secondary) schools in England were required to teach the new national computing curriculum. The curriculum included topics such as understanding and applying the fundamental principles of computer science, learning about how computational systems work and using technology to develop pupils' ideas to design and build their own programs. The teaching of computer programming in schools is not new, educational programming languages such as Logo were named explicitly in the programme of study of secondary national curriculum documents in the 90s in mathematics. However, as Manches and Plowman (2017) highlight, the more recent discussions about teaching programming in schools often fails to include the earlier wealth of research conducted during this time, reflecting the challenge of implementing research findings to affect practice. For example, the large body of research from the six Logo and Mathematics Education (LME) conferences running from 1985-1992 captured in Hoyles and Noss (1992), is often overlooked in newer research which uses Scratch programming.

The 2013 computer curriculum opens with a purpose of study overview statement to highlight the connection of computing to mathematics:

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. (Department for Education, 2013b, p. 178)
However, the curriculum document does not specify how such links should be made or the role of the teacher in achieving this purpose. The potential benefit of developing mathematical thinking skills through learning to program has been the subject of research for several decades (Du Boulay, 1980; Hoyles and Noss, 1992; Hoyles et al., 1991; Hoyles and Sutherland, 1989). There is evidence that programming can benefit the learning of specific areas of mathematics. The summaries of Logo research (Clements, 1987; Clements, 2000; Clements, 2002; Clements and Meredith, 1993; Clements and Sarama, 1997; Yelland, 1995) identified that Logo can support the learning and understanding of certain geometric concepts such as the notion of turn, or concepts of algebra and ratio and proportion. However, researchers have struggled to agree whether programming benefits mathematical understanding, largely due to the critical role of the teacher in their student interventions. This raises the question what are the skills needed by the teacher to support their students learning at the intersection of mathematics and computer programming? Given this is a new activity in primary schools, it would therefore follow that most teachers would not have the knowledge or an appropriate skillset to teach this new technical subject and to exploit the understanding developed to promote mathematical learning. Thus, there is a need to research the evolution of teachers’ knowledge as they engage with learning programming to explore mathematics.

1.2  Personal motivations for the study

I first encountered Logo programming in 1993 during my initial teacher training. I was immediately drawn to the simplicity of typing commands to control the turtle on the screen and to explore mathematical concepts. The elegance of creating a program to draw a square which could easily be modified to tile the screen was both satisfying and intriguing. When I qualified and began teaching, I regularly taught Logo to lower secondary school children through an eight-lesson project. The project included activities which introduced simple procedures to control the turtle, an exploration of external angle of polygons as well as tesselations. The children enjoyed working with Logo and were surprised when they discovered that the total angle turned for one trip around all polygons was three hundred and sixty degrees. Returning to academic study in 2010 after ten years within the technology industry, I was pleased to
discover in an early Masters’ lecture that technology usage within the classroom sometimes still used Logo but more frequently featured dynamic geometry systems (DGS). As a student researcher I had the opportunity to explore children’s mathematics meanings developed as they explored quadrilaterals both within Logo and DGS microworlds and to identify what was potentially lost due to the decline of Logo usage within the mathematics classroom.

Returning to the classroom as a secondary mathematics teacher after academic study in 2011 I continued to use Logo with my students, when I discovered that a computer room was available, but the mathematics department rarely used it. I was surprised to find that the children’s interest in Logo had endured despite Logo’s simplistic graphics compared to other graphically rich software that children had access to. My year 9 low attaining group were amazed at the simple yet colourful line art Logo polygon patterns they could easily create. In 2013, Scratch 2.0 was released, a visual block based programming based on Logo. The new version featured an overhaul of the interface which resulted in a more attractive presentation, especially compared to the older style of the Logo interface. In 2014, as part of a consultancy project where I built online Scratch applications which were used in a free online course Citizen Maths\textsuperscript{1}. Through this experience, I was excited by the general accessibility and advantages of using Scratch over Logo due to its dynamic graphics, attractive blocks-based tangible interface and the vibrant online community.

In my current role working as a lecturer in Mathematics education, I teach, and support student teachers enrolled on secondary mathematics initial teacher education courses. I regularly use Scratch to demonstrate how programming can be used to explore mathematical ideas. In a session I introduced Scratch to explore drawing regular polygons to make a simple house consisting of a square with an equilateral triangle roof on top. The student teachers had been educated within the technical age where computers were ubiquitous in their use both in and out of the classroom. I had therefore expected the teachers to be able to complete the Scratch task without issue. However, I was surprised to observe that the student teachers struggled when engaging with the angle concepts needed to calculate the angle of turn to draw the roof on the

\textsuperscript{1} Citizen Maths was a free, online course developed to help improve grasp of maths through five powerful ideas for adult learners who had previously struggled with school mathematics.
house. That is the same task which had been researched with children twenty five years ago, where children had used the angle of turn as 45 degrees to represent 60 degrees (Hoyles and Sutherland, 1986). Many of the student teachers also found the computational ideas of loops and expressing generality in mathematics and in programming challenging. For example, to create a procedure to draw any regular polygon of any size. The observations supported my assumption that the introduction of computer programming as required by a new curriculum would require both carefully designed activities and teacher professional development to teach mathematics through programming.

This research, therefore, seeks to explore three broad areas when teaching carefully designed mathematical activities through programming:

   i) What knowledge teachers need to know in mathematics and in computing,
   ii) What teachers need to know how to do, and,
   iii) The evolution of teachers’ mathematical knowledge as they engage with mathematics through programming.

The teachers for this study are selected from those that were involved with the ScratchMaths project, and the carefully designed activities are taken from the ScratchMaths curriculum (UCL ScratchMaths, 2017).

1.3 Introducing the ScratchMaths project

The ScratchMaths (SM) project was a three-year research project (2014-2017) involving the development of i) a two-year computing and mathematics curriculum, ii) professional development (PD), and iii) a two-year national classroom intervention for children aged nine to eleven years old. Within the English school system, children aged nine and ten are in Year 5 of primary school and children aged ten and eleven are in Year 6, the final year of primary school. The intervention was conducted as an independently evaluated randomised control trial within the Education Endowment Foundation (EEF) project framework. For three years I worked as part of an interdisciplinary UCL team taken from mathematics education and computer science who developed the ScratchMaths curriculum and professional development
In the second year of the intervention, I worked closely with Professor Ivan Kalas, who provided expertise in computer science and Scratch programming, to plan, design and test the Year 6 curriculum materials which had an explicit mathematical focus. Additionally, working with Dr Laura Benton, I planned and facilitated the national professional development for the second year of the intervention. The curriculum was comprised of approximately twenty hours teaching time across each of the school years and organised into three modules per school year. The first year of the curriculum in Year 5 focuses on computational thinking with an implicit mathematical component and the second year of the curriculum in Year 6 foregrounds mathematical concepts using the programming skills developed within the first year.

The curriculum materials were developed through cycles of iterative design research using a series of design workshops utilising London primary school teachers to act as ‘design partners’ (Benton et al., 2016). The design phase established that the curriculum materials needed to be appropriate for all teachers including those who had no previous experience of using Scratch or programming. The content of the curriculum was developed from a series of intensive internal design workshops to map out a high-level overview of the two-year curriculum, influenced by the team’s prior research and experiences. For example, my professional experience of teaching using Logo in the classroom and from my mathematics education experience of teaching mathematical concepts when working with beginner teachers, informed the design of the mathematical content for the Year 6 modules. Each module is sequenced into a series of four investigations, each containing up to four activities using either the computer or off the computer, known as unplugged activities. The mathematical topics for the Year 6 curriculum were organised into three modules, i) Module 4 - Building with numbers, ii) Module 5 – Exploring mathematical relationships and iii) Module 6 – Coordinates and geometry.

Activities from Module 4 and from Module 5 were selected from the start of the Year 6 curriculum for research within this study. Each module uses the computational concepts developed in the first three modules of the curriculum studied in Year 5 to explore mathematical ideas. Module 4 focuses on building and using place value models, a key topic since place value is taught in every year of the primary school mathematics curriculum. Module 5 introduces the idea of variable in computing and how it is used to express generality and relationships in mathematics by drawing polygon shapes. Each module comprises a set of
curriculum resources which included i) a teacher guide designed to give step by step activity instructions and to provide additional support and examples, ii) unplugged printable worksheets to be completed away from the computer iii) classroom PowerPoint presentations designed for display on the interactive whiteboard and contained activity instructions and discussion questions, iii) Scratch starter projects which had been carefully developed to include specifically chosen sprite(s) and pre-prepared blocks and scripts.

Each module uses a separate microworld, “an environment where people can explore and learn from what they receive back from the computer in return for their exploration.” (Hoyles, Noss and Adamson, 2002, p. 30). The idea of microworld can be viewed from two perspectives i) as a concrete embodiment of a mathematical structure that is also extensible, e.g., objects can be combined to build new ones and ii) as transparent, so that the inner workings are visible through different representations. Learners engage with the microworld as both the user of the microworld and also a designer within the microworld. These twin roles for the learner align with the theoretical idea of Papert’s constructionism. Constructionism shares constructivism’s view of learning as building knowledge structures through internalisation of actions, but goes on to add the idea that this happens “especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it’s a sand castle on the beach or a theory of the universe” (Papert, 1991, p. 1). The public entities created for the purposes of this study are computer programs or scripts built to explore and represent mathematical ideas in the context of a microworld. The ScratchMaths microworlds are discussed in detail in Chapter 4.

The teacher materials were developed to incorporate a pedagogical framework, named the 5Es, consisting of five unordered constructs, Explore, Explain, Envisage, Exchange, and bridge (see Chapter 2), based on research into good practice in teaching mathematics. The 5E’s are written explicitly into the learning objectives, the step-by-step activity instructions and in the mathematics connections column of the ScratchMaths Teacher Guide (see Chapter 4). The intervention teachers, as the participants of this study, were introduced to the 5E pedagogical framework and the curriculum materials in a series of two-day PD activities that began at the end of Year 5 before the teachers started to teach the curriculum in Year 6. I facilitated the
Year 6 PD sessions and worked with fifty primary teachers in the areas of London, Blackburn, Bradford, Merseyside, Somerset and Stafford. During the two-day PD session I observed differences in how the teachers engaged with the ScratchMaths activities. Some teachers found the programming aspects challenging, others found bridging to mathematics difficult whilst others found the programming easy and were able to explain their solutions confidently. Amongst the teachers who attended the PD, some had taught the Year 5 ScratchMaths curriculum the previous year. Other teachers had no previous experience of Scratch or programming but were expected to teach ScratchMaths at the start of Year 6. Monaghan (2004) has argued that the integration of technology into classroom practice is a complex undertaking. He explained that the teacher must adapt to managing several kinds of activity in the classroom. The variation of starting point for each of the teachers would therefore provide a unique research opportunity to trace the evolution of their knowledge as they engaged with the ScratchMaths curriculum to teach mathematics through programming.

1.4 Structure of the thesis

In Chapter 2, I present a discussion of the literature appropriate to this study. The review starts with an examination of the literature related to mathematical thinking and reasoning and then narrows the focus to research on the specific mathematical topics of place value and variable, the mathematical foci of ScratchMaths modules 4 and 5 respectively. I then introduce using computer programming to teach mathematics in the form of Logo and Scratch and conceptualisations of angle and variable using computer programming. I then discuss the research focus of the practices of teachers through an examination of their developing knowledge and discuss different knowledge frameworks. I explore the relationship between computational thinking and mathematical thinking and propose a conceptual framework which brings together the domains of knowledge required for teaching mathematics through programming. Finally, I present the research questions I aim to address in my doctoral research.

In Chapter 3, I explain the methodological approach of the research. The chapter introduces the research design, the selected methods for data collection and the participants of the study. I then introduce my analytical approach for data analysis and detail each step of the analysis.
process. I discuss how I modified and used Schoenfeld’s (2018) Teaching for Robust Understanding (TRU) framework as an analytical tool for lesson observation analysis. Finally, I consider and discuss ethical issues.

In Chapter 4, I discuss the ScratchMaths curriculum including the design of the *Place value*, and *Polygon fireworks (variable)* microworlds, which make explicit the mathematical and computational knowledge embedded within the ScratchMaths curriculum activities taught by the participant teachers in their ScratchMaths lessons.

In Chapters 5, 6 and 7, I present the results for each individual case teacher and describe and analyse how they used Scratch and the ScratchMaths curriculum in their classroom practice. Results for each case are presented as three themes organised on the conceptual framework of knowledge for teaching mathematics through programming i) Window on mathematical knowledge for teaching, ii) How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum, and iii) Teacher pedagogic practices.

Chapter 8 follows a similar structure but presents teacher episodes from two different groups of teachers’ practices who were observed less frequently compared to those in the previous chapter.

Chapter 9 reports an overview of the key findings emerging from the data analysis of the PhD research, followed by a general discussion.

Finally, Chapter 10 concludes the thesis and details my contribution from the perspective of the findings and from the perspective of methodology. I discuss the limitations of the findings and the opportunities for further study.

Note: Appendix I includes an overview and glossary of Scratch and Scratch terminology used throughout this thesis.
Chapter 2  Literature review

2.1  Introduction

This chapter provides a review of the background literature and a synthesis of the research findings from empirical and theoretical research relevant to this study. The literature review’s purpose is to develop a conceptual framework to situate the research, to identify any theoretical gaps and to refine the research questions. The review is organised into three parts; i) what is known about the teaching and learning of the specific mathematical topics in focus within this study and why they are important (section 2.2), ii) the role of computer programming to learn mathematics (section 2.3), and iii) the study of teachers practices via teacher knowledge and a discussion of frameworks for analysing their developing knowledge for teaching mathematics through programming (section 2.4).

In the first part, I review the literature on mathematical thinking and reasoning and narrow the focus to research related to the specific mathematical topics of the ScratchMaths curriculum Modules 4 and 5, place value and variable. In the second part, I introduce how programming has been harnessed to teach mathematics and examine the similarities and differences of variable represented in mathematics and in programming. ScratchMaths Module 5 - Exploring mathematical relationships, uses the vehicle of Papert’s (1980) “Turtle Geometry” to explore the relationships between variable quantities by drawing regular polygon firework patterns (see Chapter 4 for more detail). Although angle concepts are not explicitly taught within this module, teaching using the ScratchMaths microworld provides a secondary research opportunity to examine teachers’ knowledge of angle. I therefore summarise literature related to the understanding and difficulties related to the concept of angle. I then introduce the notion of computational thinking and review the research findings to explore the relationship between computational thinking and mathematical thinking. Finally, I outline the knowledge
frameworks that are used within this thesis to analyse teacher practices via teacher knowledge and detail the resulting conceptual framework. Thus, as a result of the literature review, I seek to identify why place value, variable and angle are important and hard mathematical topics to teach and learn and then consider what knowledge might be required to teach these topics effectively through programming.

2.2 Mathematical foci of the study: mathematical thinking and reasoning, place value and variable

2.2.1 Research on the teaching and learning of mathematical thinking and reasoning

Over the past twenty-five years Mathematics curricula have undergone considerable reform driven by political, societal and developmental goals aiming that children are prepared for the workplace demands of the 21st Century. There have been many exhortations to refocus the curriculum on elaborating mathematical reasoning and how students might learn to reason mathematically. There is consensus among the UK in the current National curriculum (Department for Education, 2013a), the USA (National Council of Teachers of Mathematics, 2000) and Singapore (Ministry of Education, 2000) that the aims of mathematics go beyond being able to perform taught processes and procedures. Yackel and Hanna (2003) state that mathematics educators and mathematicians use the term mathematical reasoning without clarification or elaboration. They go on to discuss that one approach is to view mathematics as reasoning contrasted with the view of mathematics as following rules and procedures. This distinction is analogous to Skemp’s (1976) seminal article on instrumental and relational understanding. Although Skemp’s categorisation is useful he did not develop the constructs further or provide any specification of how to develop either type of understanding.

More recently researchers Ball and Bass (2003) have connected reasoning with understanding by arguing that mathematics reasoning is a basic skill and “that mathematical understanding is meaningless without a serious emphasis on reasoning” (Ball and Bass, 2003, p. 28). They argued that knowing mathematical ideas and procedures as fact or routine is not sufficient to
use the ideas flexibly in other situations, nor is it helpful when reconstructing faded knowledge should the need arise. It follows that if reasoning is a basic skill of knowing and using mathematics, the work of teaching must include helping children learn to engage in mathematical reasoning beyond just being able to perform procedural operations. When mathematical reasoning and problem solving are brought into focus within teachers’ knowledge it necessitates a discussion of what it means to think mathematically and how to develop *mathematical thinking* (MT). Burton (1984) and later Schoenfeld (1992) took the perspective that to think mathematically means to develop a mathematical point of view in which the processes of matematization and abstraction are valued and applied naturally. Stacey, Mason and Burton (1982) consequently identified four fundamental processes organized as two pairs to show how thinking mathematically very often follows by alternating between them:

- Specializing and generalizing – trying special cases, looking at examples and looking for patterns and relationships,
- Conjecturing and convincing – predicting relationships and results and, finding and communicating reasons why something is true.

They argued that although specific results are useful it is the general result which is characteristically mathematical. The categories were exemplified through teachers use of questions and prompts as in later research from Watson and Mason (1998).

A complementary approach to defining mathematical thinking was taken by Cuoco, Goldenberg and Mark (1996) who also place a greater focus on the general and draw attention to the processes of mathematicians rather than specific mathematical results. They describe these processes as the *habits of mind* that are used by the mathematicians who create those results. Thus, the development of mathematical habits of mind allows learners to develop and to adopt some of the ways that mathematicians think about problems. Cuoco, Goldenberg and Mark proposed that the habits of mind can be used as an organizer for mathematics curriculum and that learning mathematics should be an opportunity for learners to be pattern sniffers, experimenters, describers, tinkerers, inventors, visualizers, conjecturers and guessers. In this
study I have adopted Cuoco, Goldenberg and Mark’s definition of mathematical thinking since the eight habits of mind have connections with both the ScratchMaths 5E pedagogical framework and the notion of computational thinking; I discuss this mapping in section 2.3.6 below. However, in the next section I narrow the mathematical focus to the specific mathematical content areas of place value and variable.

2.2.2 Research on the teaching and learning of place value in mathematics

Place value is a fundamental concept of whole number arithmetic and is a core component to most school mathematics curricula. In the National Curriculum for Mathematics in England, it is a statutory requirement that in each year of primary school every child must be taught place value concepts (Department for Education, 2013a). Looking at the history of place value, Dehaene (1992) described different human number notations and how they have passed through several stages of increasing efficiency as place value systems have evolved. The simplest numerical systems use a number of similar tokens to make a one-to-one correspondence with an organised list of number names, such as four vertical lines to represent 4 in Egyptian hieroglyphic notation. The numerical system evolved through the use of additive notations to solve the problem of memorising new symbols for each number. The Sumerians attributed special symbols to 10, 60, 100, 600, 3,600 and 36,000, fundamental numbers which underpin some mathematical concepts today e.g. 60 seconds in a minute, 60 minutes in an hour (Menninger, 1969).

The place value mathematical system

The numerals 1 through 9 of the place value system in use today first appeared in India in inscriptions from the third century B.C. (Fosnot and Dolk, 2001). The development of zero, an essential concept in order to record the absence of a place, and the combination of positional notation (around 500 AD) produced a powerful new system of notation which spread throughout Europe (Haylock and Manning, 2019). For example, to represent the number 377, the written number would contain two seven numerals, but each of the two sevens represent different quantities. Reading from left to right, the first seven represents seven tens and the second seven represents seven ones because of the places in which they are written. It is likely
that the invention and use of a positional system for writing integers and decimals is derived from its computational power (Menninger 1969). Algorithms that use numbers expressed in this way can be reduced to performing calculations based on digits only as in the written algorithm for multi digit addition and subtraction.

The place value system in the decimal Hindu–Arabic system is characterised by two properties (Houdement and Tempier, 2019; Ross, 1989):

- The quantity represented by the individual digits are determined by the position they hold and corresponds to a unit. For example, thousands are in the fourth place to the left of the decimal point. This is known as the positional property.
- The values of the position increase in powers of ten from right to left. Each base ten unit is equal to ten units of the immediately lower unit (for example one thousand = ten hundreds), i.e., the base ten units are in a multiplicative relationship with each other. This is known as the base ten or decimal property.

The mathematical beauty of the place value system lies in its structure as each number in base ten is implicitly a sum, that is we can write 1972 in expanded form:

\[ 1972 = 1000 + 900 + 70 + 2. \]

Surprisingly, in the early mathematics education place value literature (1969-2005) there was no common or standard name for the numbers within the sum until Beckmann (2008) introduced the term place value parts. These place value parts also contain a multiplicative structure in that the digit records the number of copies of the base ten unit used to create that number. Continuing the previous example:

\[ 1000 = 1 \times 1000, 900 = 9 \times 100, 70 = 7 \times 10 \text{ and } 2 = 2 \times 1 \]

The place value system is therefore comprised of multiple mathematical relationships which together underpin its structure; a number is the sum of place value parts, each place value part is a single-digit number multiplied by a base ten unit and each base ten unit is indicated by the position of the digit in the base ten expression. Howe (2019) characterised the relationships as forming a series of five stages and argued that for the purposes of mathematics teaching it is important to make each of them explicit:
Stage one is the standard form of writing numbers, stage two identifies the place value parts and expresses the number as a sum (expanded form), stage three refines stage two and makes explicit the multiplicative relationship of base ten unit and the digit within each place value part essential for a conceptual understanding, defined as knowledge beyond processes and able to explain the rationale underlying the arithmetic algorithms (Ma, 2010) for base ten addition and subtraction. The last two stages are further refinements of stage three making more explicit the multiplicative structure of the place value parts; the fourth stage expresses the base ten units as powers of ten, and the final stage use exponential notation to write the powers more compactly. The understanding of the place value system, as exemplified by Howe’s stages above is recognized by the research community as fundamental to grasping mathematics (Bartolini Bussi and Sun, 2018). Deep understanding of the place value system must therefore be essential knowledge in teachers’ mathematical knowledge for teaching.

Understanding place value

In the late 1970s and 1980s in England a national study, the Assessment of Performance Unit (APU) researched methods of assessing and monitoring achievement of students at aged 11 in primary school and aged 15 in secondary school. The research found that 86% of 11-year-olds (Foxman et al., 1980) could correctly select the number worth seven tens from a list of different numbers which might suggest a good understanding of place value. However, interviews of 11-year olds (Foxman et al., 1982) found only 36% of children could answer correctly that the value of the cubes had increased by ten as they moved from the ones column to the tens on a place value notation board. The same result was found within the large-scale Concepts in Secondary Mathematics and Science study (Hart et al., 1981). These findings led researchers to explore children’s difficulties in learning place-value concepts (Fuson and Briars, 1990; Fuson et al., 1997; Hart et al., 1981; Hiebert, 1997; Hiebert and Wearne, 1996; Kamii, 1986; Ross, 1989).
Kamii (1986) found that children do not have trouble putting out the correct number of counters when they are shown the numeral 16 and have little difficulty in writing the numeral when they are shown sixteen counters. Kamii argued that both tasks can be solved by counting and memorising ordered number words or numerals. However, the difficulty arises when there is part-whole relationship of the digits involved in the notation, e.g., the value of the 1 in 16. Fuson et al. (1997) proposed that children move through different stages as their conceptions of two-digit whole number place-value develops. They explained that each stage involves a triad of two-way relationships between number words, written number marks, and quantities. Within the stages, several different conceptions may be held by a child that can be used in different contexts. However, an incorrect conception can also develop, a misconception that can exist alongside any of the other correct conceptualizations. For example some children interpret the digits in 72 as independent digits consisting of a seven and a two, a concatenated single-digit conception (Fuson et al., 1997). This conception is particularly seductive when children are faced with problems represented vertically, despite having one of the correct multidigit conceptions available for correctly adding or subtracting numbers when solving word problems or problems presented horizontally (Fuson and Briars, 1990). This is challenging for teachers given the importance of pencil and paper written algorithms in the KS2 national mathematics tests.

Another important finding from the place value research is that some children may develop a face value (Ross, 1989) conception of place-value. This conception is formed where each digit represents the number indicated by its face value and therefore the set of objects represented by the tens digit may be different from the set of objects represented by the ones digit. Ross found that a child may verbally label the tens digit as “tens” without grasping that they truly represent groups of tens units and therefore do not recognise that the number represented by the tens digit is a multiple of ten. This misconception is likely to occur when children use pre-grouped materials such as base ten blocks without careful teacher intervention since the child does not have to think about how big each of the blocks are. Despite the wealth of research into children’s understanding of place value, poor success rates continue in tasks involving relations between units (Houdement and Tempier, 2019).
Teaching place value concepts and teachers’ knowledge of place value

The teaching and learning of place value concepts continue to be a focus for researchers in mathematics education (see for example the 23rd International Commission on Mathematical Instruction (ICMI) study, Building the Foundation: Whole Numbers in the Primary Grades, Bartolini Bussi and Sun (2018)). Seminal research from (Ma, 2010) has informed studies which explore teachers’ knowledge for elementary mathematics. Her initial study compared Chinese and U.S. elementary school teachers and found distinct differences in how each group construed elementary mathematics in their teaching. The U.S. teachers considered that elementary mathematics was basic, and an arbitrary set of facts and rules, in which doing mathematics means to follow step by step procedures to arrive at answers (Ball, 1991). However, the Chinese teachers were concerned with knowing why algorithms worked such as in explaining borrowing or carrying in the algorithm for multidigit subtraction or addition. The difference between the Chinese and the U.S. teachers is particularly significant given China’s high performance on the Trends in International Mathematics and Science Study (TIMSS, 2019) assessment. Ma’s finding suggested there is a need to reconsider teachers’ knowledge for teaching place value concepts.

Researchers such as Thanheiser (2009) explored preservice elementary U.S. teachers’ conceptions of multidigit whole numbers. They found that teachers’ familiarity with the base ten system, and their lack of an understanding of multidigit whole numbers, led to teaching that encouraged a superficial understanding of the system that relied on recitation of the value of the digit in the ten’s place. Using this finding as a starting point, Cady, Hopkins and Price (2014) designed a study which used an invented number system of base which focused on the mathematical place value ideas of symbols, position and base. They found that teachers gained an understanding of counting and grouping by five (the base of the invented Orpda number system) and used this knowledge to reflect upon their own limited background knowledge of place value. The authors concluded that engaging teachers in activities which support teachers to reconceptualise their understanding of seemingly simple concepts would be an effective way to engage with place value knowledge they took for granted.
2.2 Mathematical foci of the study: mathematical thinking and reasoning, place value and variable

A different approach to develop teachers’ knowledge was taken by Venkat (2015) who researched primary teachers use of key representations in South America. She found that a focus on key representations, such as double number lines and ratio tables, introduced and discussed in the context of whole number arithmetic, appeared to be an important component that simultaneously supported the development of the teachers’ mathematical learning and of their mathematics teaching. The ICMI 23 (Bartolini Bussi and Sun, 2018) group considered that this finding is of particular interest to the literature based on primary mathematics teacher knowledge since it supports the requirement of an additional specialized type of knowledge for mathematics teaching. Although Venkat’s work was not in the specific context of place value, the finding provides further evidence of the importance of representations in the development of mathematical knowledge for teaching.

A key representation in the teaching of place value has been the availability of base ten blocks or Diene’s blocks in school classrooms following the work of Hungarian researcher Dienes (1963). Research has found that base ten blocks are useful when demonstrating an abstract mathematical concept of the number system such as one to one correspondence (the ability to match an object to the corresponding number and recognise that numbers are symbols to represent a quantity) or to improve children’s explanations of why the algorithm works for arithmetic place value operations (Carpenter et al., 1999; Fuson and Briars, 1990). However, using the base ten blocks might not always be helpful to young children if a teacher does not understand the mathematical meaning of the representation (Green, Piel and Flowers, 2008). Walkerdine’s (1988) research on place value lessons involved children working on activities to group matchsticks into groups of ten. Her observations highlighted a central aspect of the teacher’s role in manufacturing a connection between the value of the numeral and the pile of math sticks through the teacher’s linguistic acts. Walkerdine noted that the teacher hoped that this correspondence would be obvious to the children, without making the correspondence explicit to the children. Cobb et al. (1992) critiqued the study arguing that in the absence of any analysis of the children’s contributions of their interpretations, they can only hope that like the teacher the place value correspondence was obvious to the children.
The importance of the role of the teacher to make connections within mathematics is well understood in helping students develop conceptual or deep understanding. See for example Askew et al. (1997) or for a more comprehensive review see Walshaw and Anthony (2008). Making connections must therefore be an important factor in teachers’ pedagogical approaches when teaching with place value artefacts. The importance of the teacher to make connections has also been identified in the use of technology such as virtual manipulatives to mediate the teaching and learning of place value concepts. Sarama and Clements (2009) in their review of several studies reported that the physicality of the manipulative itself is not important for meaningful learning, but the usefulness of the manipulative comes through its use in well planned and instructive contexts which involve the teacher. Other studies have used the benefits of multi-touch technology to introduce new possibilities in the design of virtual artefacts, such as TouchCounts an iPad application developed by Nathalie Sinclair. The application provides an environment for children to use their fingers, eyes and ears to learn to count, add and subtract. The approach draws on the extensive neuroscientific literature connecting use of fingers with the development of number sense, “With multitouch technology, the interaction becomes more immediate, as the fingers contact the screen directly, either through tapping or a wide variety of gestures.” (Sinclair and Baccaglini-Frank, 2016, p. 670). The potential for use of technology to explore place value topics is enticing, however, Butterworth (2018) warns that the design of new technology approaches must do more than rehearse students in what they already know. He argues that any software developed should draw on established pedagogical principles such as in the use of constructionism.

In summary, the place value literature suggests that knowledge of children’s conceptual problems of understanding the place value system is stable yet continues to be a challenge for teaching and learning. The role of the teacher is critical to be able to make connections within mathematics, including the use of representations to help children to develop deep understanding. Some studies have shown that teachers own knowledge of place value can limit their capacity to make such connections and that existing teaching approaches and content may be problematic since they do not allow the teacher to confront their existing knowledge and to reconceptualise it if required. Other studies have found the dangers of focusing on only using rote drill of written procedures provides students with little chance to construct relations
2.2 Mathematical foci of the study: mathematical thinking and reasoning, place value and variable

between these procedures and other things they know (Hiebert and Wearne, 1996, p. 254). Using technology has provided new ways to engage with place value knowledge not possible with pencil and paper methods or physical apparatus. Together, the research findings suggest a rethink of the approach to teaching place value is required and that a constructionist practice such as computer programming may provide new opportunities for teaching and learning place value concepts.

2.2.3 Research on the teaching and learning of algebraic variable in mathematics

Over the past 40 years, research has addressed many of the problems associated with the teaching and learning of algebra making it one of the most researched areas in mathematics education. In the plenary of the 6th congress of the European Society for Research in Mathematics Education (CERME), Radford (2009) posed the question as to whether there was anything new to say about algebraic thinking. The working group positively responded “Yes!” as their papers demonstrated that there was not yet a consensus on some of the fundamental questions associated with the teaching of algebra such as mathematical task design or how general or specific teaching using technology should be (Hodgen, Oldenburg and Strømskag, 2018). The lack of consensus provides a research opportunity both in task design and in the use of technology of this study. Unlike place value which the National curriculum prescribes to be taught in all years of primary education, the topic of Algebra is only explicitly stated within the Year 6 programme of study of the National Mathematics curriculum. In Year 6, the National curriculum expects teachers to introduce the use of symbols and letters to represent variables and unknowns in mathematical situations such as missing numbers, lengths, formulae in mathematics and science, generalisations of number patterns and simple number puzzles (Department for Education, 2013a, p. 43).

Early algebra is a relatively new field of research which has a focus on younger students aged between 6 and 12 years old (Kieran, 2018). Mathematical relationships, patterns, arithmetic structures alongside the mathematical thinking processes of noticing, conjecturing,
generalising, representing are central to early algebraic activity. There have been numerous studies that have discussed children’s difficulties in moving from an arithmetic to an algebraic form of reasoning (see for example Kieran, 1992; Linchevski, 1995; Rojano and Sutherland, 2001). The difficulties provided a stimulus to explore whether certain kinds of algebraic activity, broadly referred to as algebraic thinking, might be accessible to younger children and support the transition into a more formal study of algebra.

Kaput (2008) distinguished between algebra as a self-standing body of knowledge embedded in educational systems i.e., a cultural artifact and algebraic thinking as a human activity, an activity from which algebra emerges. He explained that algebraic thinking has two core aspects, 1) “generalisation and expression of generalisations in increasingly systematic, conventional symbol systems” and algebra as 2) “syntactically guided action on symbols within organised systems of symbols” (Kaput, 2008, p. 10). Further, he claimed, that the two core aspects of algebra are embodied in three strands of school algebra: i) algebra as the study of structures and systems abstracted from computations and relations including algebra as generalised arithmetic and in quantitative reasoning, ii) algebra is the study of functions, relations, and joint variation; and iii) algebra as the application of a cluster of modelling languages inside and outside of mathematics. Kaput’s distinction between algebra and algebraic thinking underpin the design of the ScratchMaths Module 5 – Exploring mathematical relationships discussed in Chapter 4.

One of the most significant steps in early algebra is understanding the change in the role of letters in mathematical expressions from unknowns to variables (Blanton et al., 2015; Carraher and Schliemann, 2007; Ely and Adams, 2012). Variables in mathematics can take on multiple meanings. Usiskin (1988) defined variable as i) generalised arithmetic as pattern generalisers, ii) unknowns or constants in the context of solving problems, ii) arguments and parameters in the context of relationships between quantities or iv) arbitrary marks on paper in the context of structures. Philipp (1992) considered the different roles of variable in mathematics such as labels, constants, unknowns, generalised numbers, varying quantities, parameters or abstract symbols. With respect to how children view variable, Kuchemann's (1978) large scale research resulted in the creation of a hierarchy of variable from the child’s perspective. The hierarchy moved from where the letter is directly evaluated e.g., $x + 5 = 12$, so $x = 7$, through different
stages where the letter represents an object, such as P for perimeter, or the letter represents a specific unknown, to contexts when a variable represents a set of numbers rather than a specific number. The highest level of the hierarchy is where the letter represents a variable, whereby the relationship can vary dependent upon the value that is represented. E.g., identifying which is larger 2x or x+2. The hierarchy therefore represents the challenge that children have with thinking of variable as a generalised number. Phillipp (1992) similarly argued that much of the difficulty children encounter with variables is their inability to recognise the correct role of the variable. The role of variable is therefore an important aspect of teacher knowledge when teaching variable concepts in mathematics.

Despite the substantive body of research on theoretical conceptualisations of the meaning of school algebra and the difficulties children face when learning of algebra, there is a lack of a substantial body of research on teachers’ knowledge and practice in teaching of algebra (Doerr, 2004). The 12th ICMI study conference (Stacey, Chick and Kendal, 2001) called for research to explore many areas of teachers’ content knowledge such as the concept of variable which had yet to be investigated. Despite this call few papers have been published to respond to this need. Brown and Bergman (2013) explored preservice primary teachers’ understanding of variable and found that many of the misconceptions of variables displayed by school children were also present in the teachers’ response to the same variable test items. Wilkie’s (2014) research of in service teachers’ understanding of function machines i.e. using a table of paired variables, found that less than half of the 105 teachers surveyed had appropriate pedagogical content knowledge. Similarly, Girit Yidiz and Akyuz’s (2019) examination of a small group of middle school teachers, found that teachers used the unknown and variable concepts as if they had the same meaning. These findings highlight the difference between knowing how to do the mathematics itself and knowing the why in order to be able to teach effectively. Dogbey (2016) suggests that teachers’ mathematical knowledge for teaching variable and related concepts is unlikely to improve without changes to the curriculum which support it. He recommends designing a curriculum which provides opportunities for teachers to be able to delve more deeply into the concepts of variable in school mathematics and to treat variables as a topic as an approach to develop the required mathematical knowledge.
In summary the literature has identified the difficulty children have to develop their understanding of variable concepts in mathematics and has provided theoretical models to support the development of variable concepts. There are limited studies on teachers’ knowledge and practice in teaching algebraic concepts and there has been a call to develop research which explores teachers understanding of variable concepts and that researching new curriculum approaches may provide such an opportunity. The findings therefore support the rationale of my research to explore the teacher knowledge required to teach a newly designed curriculum which provides opportunities to engage more deeply with the concepts of variable in mathematics through computer programming. The findings of this thesis will therefore contribute to the body of research of teachers’ development of knowledge for teaching about variable concepts.

2.3 Teaching and learning mathematics through programming

2.3.1 Introducing Logo and representing mathematics through programming

In 1969 Feurzeig and Papert wrote about the challenge for most mathematics students in school: “the formal methods of mathematics remain forever mysterious, artificial, poorly motivated, and very obscurely related to intuitive thinking” (Feurzeig and Papert, 1969, p. 3). Their solution was to put forth the idea of a programming language which would impose the need of precision and rigour upon mathematical work. They proposed that computers could be used for learning about mathematics by actually doing it. Papert (1972) considered the idea that if we thought of learning mathematics like that of learning French, e.g., to learn French most effectively, you go to France, in order to learn Mathematics most effectively you go to Math-land. He described a culture of mathematics, that was analogous to that of creating soap sculptures in an art classroom. The object that the child creates is enjoyed by the teacher and is an object of discussion within the classroom. The child has an emotional attachment to the object, a sense of ownership in wanting to create, express and construct an idea in the public space. The vision led to the creation of Logo, a programming language built upon the construct that by learning to program a computer, children could explore mathematical ideas.
2.3 Teaching and learning mathematics through programming

Logo initially consisted of a turtle; a floor robot controllable by a child via a keyboard like device. As technology developed the floor robot was replaced by a screen representation of a turtle made of light, i.e., displayed on the screen as a graphical object. The turtle is controllable via the child’s commands and becomes an object to think with (Papert, 1980). He explained that a child has an intuitive understanding of how their body moves, of the geometry of their body and can use this knowledge to command the turtle. For example, a child might “Play turtle” and say “if I’m heading in this direction, and I want to go in that direction, I need to turn right by 90 degrees”. Papert defined this process as body-syntonicity to refer to experiences that are related to one's knowledge and sense about one's body as the learner engages with a different type of geometry than that of Euclid, Turtle Geometry. Turtle Geometry as expressed in Logo is dynamic and coordinate free unlike Euclid’s static point. Turtle Geometry consists of a turtle who only knows where it is, in which direction it is facing and its step size. Papert’s notion of the turtle as an object to think with and playing turtle are important ideas which underpin the design of the ScratchMaths curriculum materials discussed in Chapter 4.

The publication of Mindstorms (Papert, 1980) provoked an interest in researching the effects of learning to program the computer on children’s understanding of mathematic concepts. Since that time global researchers have documented over 30 years of Logo research. Children engaging with Logo within mathematics classrooms were commonplace in the 80’s and early 90’s and featured specifically in some curricula. Logo and its numerous iterations were used to research and explore the breadth of the mathematical curriculum from working with very young children to students at university. There have been several summaries of Logo research (Clements, 1987; Clements, 2000; Clements, 2002; Clements and Meredith, 1993; Clements and Sarama, 1997; Yelland, 1995). The major themes that are important for this study are that programming in Logo can support the learning and understanding of some mathematical concepts such as the notion of turn and algebraic concepts such as variable, see for example (Hoyles, Sutherland and Evans, 1986; Noss, 1985; Sutherland, 1987; Sutherland, 1992).

Despite the potential for learning mathematics through Logo there were also criticisms which took aim at what the computer could teach the student by virtue of programming. For example Pea and Kurland’s (1984) experimental study researched two classrooms in a Manhattan
private school to test whether the rigor of learning to computer program using Logo, transferred to the development of thinking and problem solving skills in other areas. The test results showed no differences in a paper-based planning task between the students who had used logo to program the computer and those who did not. This led the authors to conclude that there was no benefit to students planning skills when learning Logo programming and caused them to question whether Logo supported other kinds of cognitive development. The Pea and Kurland study was critiqued from many perspectives, most critically at the time by Papert (1987) who described Pea and Kurland as technocentric in that they saw Logo as an entity which could form a treatment while keeping everything else fixed, for example without considering the role of the teacher. Noss and Hoyles (1996) in their detailed critique of Pea and Kurland’s work some ten years later also called into further question the research methodology and analysis of results. They wrote that the research had a clear political agenda, and that the researcher held several prior assumptions about programming transfer and more importantly for the purposes of this study that learning Logo involved little or no intervention from the teacher. Unfortunately, the impact of Pea and Kurland’s study was felt throughout the community and interest in the role of computer programming began to wane. This is evidenced most clearly with the gradual removal of Logo programming from successive iterations of the National mathematics curriculum in England during the 90s. However, the introduction of Computing as a National Curriculum subject in 2012 provides opportunities to re-engage with the teaching and learning of programming.

School use of digital technology in the 80s and 90s involved overcoming additional hurdles, for example booking time in a computer room or the technical challenges such as the use of correct syntax with programming languages. However, this period also represented a significant shift in how computers were viewed in educational contexts. An advantage of using digital environments is the potential to present multiple representations, often simultaneously. Goldin (2014, p. 409) defined mathematical representations most commonly interpreted in education as:

visible or tangible productions – such as diagrams, number lines, graphs, arrangements of concrete objects or manipulatives, physical models, mathematical expressions, formulas and equations, or depictions on the screen of a computer or calculator – that encode, stand for, or embody mathematical ideas or relationship.
The ability to dynamically link and investigate visual, symbolic, aural, and numerical representations simultaneously has been documented in numerous studies, see for example (Andersen *et al.*, 2009; Goldin, 1998; Kaput, Noss and Hoyles, 2002; Morgan and Kynigos, 2014). The use of multiple representations to promote learning by highlighting different features of the representation were suggested in findings from Ainsworth, Bibby and Wood (1997). Their research of young children engaging with software which supported multi representations found that when learners connect representations with others, they have to engage in activities that promote understanding. In the context of place value, a recent study (Justin, 2013) explored virtual base ten blocks where the number represented by the base ten blocks is dynamically linked to the students’ actions on the blocks. Students used virtual blocks to accurately build quantities, write numerals, and count quantities related to place-value concepts. The representations in the virtual environment afforded students to build nonstandard forms, rename numbers that are directly linked to the multidigit algorithm, and keep a trace of actions, to support their development of place value concepts. All of these representations are challenging or impossible to achieve with the standard base ten blocks.

Another aspect of using a computer is how the computer can provide a *window* on pupil thinking (Noss and Hoyles, 1996). The window metaphor provides a heuristic that a computer can act as a window on knowledge to be looked through, not to look at. Pupils can look through the window towards mathematical objects and structure, and teachers or researchers can look through the window on pupil's thinking as ideas are constructed on screen. The computer allows a means to study *thinking-in-change* (ibid) which supposes that instead of a taking a mental snapshot of what is known, the idea is to set thinking in motion and investigate how change occurs as new notions are introduced, and the pupil makes their own connections and constructs meaning. A useful example is to consider drawing a turtle circle in Logo. While some teachers may focus on the output and the result that the child has produced, this would be missing the point, since by working in this way to draw circles can provide new ways to think about circles, and through them, new ways to think about mathematics more generally (Papert, 1987). I use the window metaphor in the analysis of teachers’ mathematical knowledge as they teach with the ScatchMaths curriculum described in Chapter 3 - Methodology.
2.3.2 Research on the teaching and learning of angle

There is no doubt that “angle is a multifaceted concept” (Mitchelmore and White, 2000, p. 209). Several authors have noted the wide variety of definitions of angle in school textbooks. Lo, Gaddis and Henderson (1996) found nine different definitions in textbooks designed for pre-service primary teachers in the USA, whilst Close (1982) found a similar finding in primary textbooks in the UK. Historically, the word angle comes from the word hook as seen in old English, Latin and Greek (Henderson and Taimina, 2005). Euclid’s definition was “the inclination to one another of two lines … which meet” (Heath, 1956, p. 176). However, angle definitions in textbooks tend to fall into three categories; as an amount of turning about a point between two lines; a pair of rays with a common end-point; and the region formed by the intersection of two half-planes.

Student difficulties in their understanding of angle is well documented in the research literature. Clements et al. (1996) found that there are two major factors which can influence students’ use of angles, i) the physical presence (or absence) of the lines which make up the arms of an angle and ii) the conceptualisation of turning as a relationship between two headings. Research from Mitchelmore and White (1998) found that students have great difficulty in coordinating the facets of angle such as turns, crossings, slopes, corners, and bent objects and paths. Their results suggested that children initially acquire a body of disconnected angle knowledge situated in a large number of everyday situations. The children then group the situations to form angle contexts such as turns and corners; and finally, form an abstract concept by recognising similarity such as static (configurational) or dynamic (related to movement) perspectives cross several contexts. Their later research highlighted that a “mature abstract angle concept” (Mitchelmore and White, 2000, p. 219) depends essentially on learning to link together the various angle perspectives in different angle contexts. In Turtle Geometry using Logo programming, angles are represented as dynamic turns rather than static direction relations, see for example Kynigos (1997) and Latsi and Kynigos (2021). Clements et al. (1996) research using Logo investigated the development of turn and turn measurement concepts in primary classrooms. They found that children gained experience with physical rotations, particularly of their own bodies, and in parallel gained limited knowledge of assigning numbers to turns. They
stated that a synthesis of the two domains of turn as body motion and turn as number constituted a critical junction for some students’ learning about turns.

Children working at a computer using Logo without teacher intervention can generate new misconceptions or compound old ones. For example, a child can create a program to control the movement of the turtle to draw the outline of a house consisting of a square and an equilateral triangle using a trial and error approach. The drawing of house is therefore constructed without appreciating how the turtle is turning and the relationship to the turn number. In classroom situations I have consistently observed that children and adults will first try 60 (degrees) as the turning angle to draw an equilateral triangle because 90 (degrees) worked to draw a square, the angle inside the shape, the same finding as earlier Logo research (Hoyles and Sutherland, 1986). The program for drawing a square hides the relationship of the turtle turn to the exterior angle of the shape since the interior and exterior angles of a square are equal. Some learners may then continue to try different turning angles rather than engaging with the perspective of angle as a number entered into Logo and how the turtle turns on screen. Whilst a completed house can both look and feel like success from the learner’s perspective, potential opportunities have been missed by the learner to engage with a rich geometric relationship.

The role of the teacher, is essential if they are to support children to develop a more mature abstract angle concepts from their work in Logo by linking those meanings to other perspectives and angle contexts (Mitchelmore and White, 2000). Some research has shown that teacher’s personal understanding of angle measure tend to be lacking in coherence and conceptual meaning (Akkoc, 2008; Thompson, Carlson and Silverman, 2007). Although notably the most recent research is related to secondary teachers’ knowledge in more advanced topics such as radians and trigonometry (see Tallman and Frank (2020) for a detailed discussion). Within the context of this study, ScratchMaths module 5 uses Turtle Geometry as a vehicle to engage with mathematical relationships including variable. Thus, angle concepts are an important aspect of teachers’ knowledge, although not taught explicitly, as they teach using the ScratchMaths microworld to explore variable concepts by drawing polygon firework
patterns. Consequently, teaching using the microworld provides a research opportunity to examine teachers’ knowledge of angle in different contexts.

2.3.3 Research on teaching and learning of variable in computer programming

Algebraic variable as discussed in section 2.2.3 can take on various meanings in mathematics such as labels, constants, unknowns, generalised numbers (Philipp, 1992). The concept of variable in programming shares similarities with the mathematical algebraic variable but also has important differences (Doukakis, Grigoriadou and Tsaganou, 2007). For example, variable in programming languages is defined as a “symbolic reference to a computer memory location that can store values of a specific type” (ibid, p. 2). However, in mathematics variables have only symbolic existence and are not stored physically as they are in computing. Variables in computing can store values of a specific type, for example, alphanumeric values. In mathematics variables use values from a number system, so in mathematics a learner would not encounter variables storing text. In equation solving, a solution can have two values, for example a quadratic curve cuts the x-axis in two places, but variables in computing can hold only one value. Finally, the values stored in a programming variable can change dynamically as a computer program is run.

In the context of Logo programming, Logo allows variables to be used in a way which models mathematical usage closely such as in using functions. For example, the Logo instruction FORWARD 50 would move the turtle forward 50 steps. The same instruction could be changed by turning the input into a variable such as FORWARD :LENGTH. Importantly, Logo unlike some other programming languages makes a clear distinction between the name of the variable LENGTH, and the value assigned to it :LENGTH by using the prefix colon, an important distinction in creating a conception of variable as generalised number. Early research from mathematics education found that Logo programming could be used to provide students with a conceptual basis for variable in mathematics (Noss, 1986; Sutherland, 1992) and that learning to program enhanced their work with paper and pencil algebra (Sutherland, 1989). Noss’s research with young children using Logo found that they developed conceptions of algebraic variable as a generalised number and that they created variables with meaningful names (almost
unheard of with introducing algebra) which suggested the potential for learning about elementary algebra through computer programming. Complementary research, from Sutherland (1992) found that Logo experiences can support early secondary school aged pupils to move from arithmetical to algebraic thinking. She showed that the pupils were able to use unknowns as a variable in Logo, operate on the unknown, and discriminate between the variables and invariants within a problem by making these relationships explicit within the formal language of Logo. Sutherland also found that pupils can make links between variable in Logo and variable in algebra, but that they depended on the nature and extent of their Logo experience. Consequently, she raises the criticality of the role of teacher in creating problems that develop pupil’s algebraic understanding and to provide information about the constraints on these ideas in the programming environment (Sutherland, 1989).

Early research from computer science education found that programmers had flawed understandings about variable. For example i) misconceptions about how variables are assigned, ii) incorrect assumptions that variables can hold multiple values at the same time and iii) understanding that an expression involving a variable inside a loop can have different values in each cycle of the loop (Du Boulay, 1986). Later research (Doukakis, Grigoriadou and Tsaganou, 2007; Sorva, 2012) supported the earlier variable findings and added that novice programmers believed that variable remembers old values rather than having the previous value overwritten, or that the previous value is still somewhere in the computer even though it is not accessible any more. Grover and Basu’s (2017) assessment of young learners found that the children were generally unfamiliar with the use of variables and harboured misconceptions about them after following an introductory course using Scratch. They also noted that learners had difficulties with other aspects of introductory programming such as how loops work. Grover and Basu argued that although Scratch had made it easier for novices to construct programs, the same misconceptions that were reported in the earlier literature remained.

Hermans et al. (2018) called into question the pedagogical approaches teacher may use to introduce variable and the misconceptions that may arise as a result. For example, a popular metaphor used in the introduction of variable is for learners to envision a variable as a box with a label on it in which a value is physically stored. This can be a nice representation since it
relates the somewhat abstract concept to that of storing something in a box. However, a box could hold one or more values, particularly when the representation of the variable uses a piece of paper which is put into the box. Their research involved assessing students’ knowledge of the misconceptions following two different approaches to teaching the concept of variable. One group were taught using the box metaphor, whereas another group were taught the explanation of a variable as a label, such as the age of a person or the temperature and then being consistent with language, x is 5, rather than x contains 5. Their findings suggest that the box group outperformed the label group when answering questions related to the assignment of one variable. However, the label group outperformed the box group when dealing with two consecutive variable assignments.

Together, the literature findings from mathematics education and from computer education point to a similar need as discussed in the research literature on the teaching and learning of algebraic variable in mathematics (section 2.2.2). That is, the topic of variable needs to be taught in meaningful programming contexts, including the need to encounter variables in loops to support deep understanding, and that consideration needs to be given to the way that teachers introduce and talk about variable. Significant to each of these factors is the teachers’ mathematical and computing knowledge of variable.

Note, in the context of using programming to explore the three mathematics concepts within this study, place value, variable and angle, there were few studies which directly explored place value concepts. The research focused more on numerical concepts, for example Kull and Carter (1990) noticed that young children used the mathematical concepts of place value and part whole relationships when pursuing Logo projects. Other authors such as Stagerz (1997) suggested the future potential for Logo to explore number theory and relationships.

2.3.4 Introducing Scratch, a blocks-based Logo programming environment

By the late 90s and early 2000s in the UK, the challenge of access to computers became less problematic as technology became cheaper and mobile laptops began to be available in some schools in addition to access to computer rooms. At the same time, it was common to find at
least one computer that the teacher or students could access in each classroom which was most frequently attached to an interactive white board. As the internet became omnipresent so did access to knowledge about computer programming, through the multitude of code sharing and discussion board websites. Developments in approaches to make programming more accessible, such as led by researchers at MIT, who had developed Logo, began to remove the barrier of typed syntax when learning to computer program by providing visual programming environments; an example is Scratch.

The Scratch project began in 2003 as a response to change the type of computer activity which took place in the after-school clubhouses set up to create opportunities for youth in low-income communities. The clubhouse vision allowed young people to work on design projects, to follow their own interests and to develop technological fluency defined as “not only knowing how to use technological tools, but also knowing how to construct things of significance with those tools.” (Resnick, Rusk and Cooke, 1998, p. 2). However, despite the clubhouse vision, Kafai, Peppler and Chiu’s (2007) study revealed little to no programming activities were taking place. Their response was the creation of Scratch a visual programming environment which allowed users to learn computer programming whilst engaged in creating and sharing animated stories and games.

Scratch is a programming environment which built upon the team’s research of “programmable bricks” commercialized as LEGO Mindstorms and grounded in the media-rich and network based activities which were popular among clubhouse youth (Resnick, Kafai and Maeda, 2005). Scratch builds upon the constructionist ideas of Logo with its easy to get started low floor or threshold, and opportunities to work with increasing complexity over time high ceiling or threshold. Scratch ads features implemented to address some of the challenges faced when using Logo. The most obvious change is the visual environment which fosters a tinkering approach since it allows the user to experiment with commands and code snippets in the same way that one might tinker with mechanical components (Maloney et al., 2010). The interface is not reliant on learning a specific syntax since the blocks snap together. The visual grammar of the block shapes and their placement and combinations replaces the role of the text based language in Logo. Other changes include the lack of error message provision and making
executing visible though visual feedback to help the user understand when scripts and triggered and how long they run. Scratch makes data concrete since it exposes variables as concrete objects the user can see and manipulate. In traditional programming languages, variables are invisible, abstract and can be difficult to understand.

In 2004, clubhouses introduced Scratch to computers, however the take up was initially slow. However, a year later a new computing culture started to emerge that to establish membership in the community, a user had to first create and share a Scratch project for others to play. This collaborative approach underpinned the Scratch platform and centred on being able to utilize someone else’s creative work and make it into a new object to use (Monroy-Hernández, 2007). Since the public launch in May 2007, the Scratch website has become an extremely active community 13 million projects shared and a large user base of over 91 million (as of December 2021), providing the ability to share, discuss and remixes users’ projects. As Scratchers program and create projects, Resnick et al. (2009, p. 80) claim that users “learn important mathematical and computational concepts, as well as how to think creatively, reason systematically and work collaboratively: all essential skills for the 21st century.”. They drew attention to the use of variable, random numbers, loops, coordinates, negative numbers, and conditional statements in an example script from a simple Pong-like paddle game. The 2013 introduction of the computing curriculum in England for all children has seen an increased use of Scratch in primary classroom settings as schools engage with the requirements of the national curriculum. The next sections examine the literature related to learning about angle and variable using computer programming which are the mathematical foci of the study.

2.3.5 Computational thinking

The past fifteen years have evidenced a renewed interest in the role of computers in education, driven by a desire to ensure that citizens are prepared for the digital demands of the 21st century. In 2006, Jeannette Wing published a seminal paper which proposed an initial definition of computational thinking (CT) as a kind of analytical thinking that should be a fundamental skill for everyone, not just computer scientists. Significant in the paper was a distinction between what CT was and what it was not stressing that CT involved “solving problems, designing systems, and understanding human behaviours, by drawing on the concepts fundamental to
2.3 Teaching and learning mathematics through programming

computer science” (Wing, 2006, p. 33). Since that time there has been considerable discussion in attempting to redefine CT. Wing’s paper served as the starting point for two National Academy of Sciences workshops where leading researchers from education, computer science department and leaders from the computing industry came together to explore “the nature of computational thinking and its cognitive and educational implications” (National Research Council, 2010, p. viii) and the pedagogical aspects of computational thinking (National Research Council, 2011). The first workshop focused on early notions of CT such as procedural thinking and programming. Although these conceptualisations were considered valid, they were revisited and broadened to encompass several core concepts of computer science beyond just programming.

Several definitions of CT followed, The Royal Society (2012) offered a succinct definition that captures the essence of CT:

Computational thinking is the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes (The Royal Society, 2012, p. 29).

Wing (2008) adapted her definition to the nuts and bolts of CT as defining abstractions, working with multiple layers of abstraction, and understanding the relationship among the different layers. In 2011, she changed her description again focussing on the “thought process involved in formulating problems and their solutions so that the solutions could be represented in a form that could be effectively carried out by an information-processing agent” (Wing, 2011, p. 1). Meanwhile, Barr, Harrison and Conery (2011), in recognising the lack of a widely accepted definition, focussed on providing an operational definition and a shared vocabulary through age-appropriate examples in order to make the concepts of CT accessible to educators. Grover and Pea (2013) traced CT back to Papert’s studies of Logo and summarised CT as having elements which included abstraction and pattern generalizations, information processing, symbol systems, debugging and efficiency.

Whilst these elements articulate the concepts for what might be considered subject content in school curricula, it does not include behaviours, practices or the social aspects of computer programming. Taking a constructionist perspective, Wolz claimed that CT was an essential
skill as reading or writing and that “programming is a language for expressing ideas. You have to learn how to read and write that language in order to be able to think that language” (National Research Council, 2010, p. 13). The public expression and social nature of computer programming within communities of practice (Lave and Wenger, 1991) are ubiquitous with using a computer and its persistent online connection. diSessa emphasizes the social construction of literacy, noting at the National Research Council 2010 conference “an effort to teach computational thinking (or computational literacy, in diSessa’s terms) to everyone is, in large part, a social problem.” (National Research Council, 2010, p. 14). Because of the wide variety of CT definitions and conceptualisations there is no widely-accepted assessment of CT. A recent synthesis of the literature (Tang et al., 2020) organised empirical studies according to those related to developing programming and computing concepts, or those related to competences needed in both domain specific knowledge and general problem solving skills. One framework which was applied to a significant number of empirical studies was Brennan and Resnick’s (2012) 3D CT framework for studying and assessing the development of CT whilst young people work with Scratch.

The 3D CT framework contains three key dimensions a) the computational concepts designers engage with as they program, such as variables, events co conditionals, b) the computational practices that developed as programmers engage with the concepts, such as working iteratively, testing and debugging, reusing and modularizing and c) the computational perspectives designers form about the world around them and about themselves, such as expressing, connecting and questioning. The framework is also significant since it has been used as the basis to define and identify expected CT skills in two recent large systematic reviews of using Scratch to learn computational thinking in compulsory school settings (Zhang and Nouri, 2019) and beyond compulsory settings (Lye and Koh, 2014). Both studies selected the framework in part due to Scratch’s theoretical foothold within the framework as well as that the framework focuses primarily on the sort of knowledge used by students as they code, rather than more general thinking skills as in Wing’s definition. Zhang and Nouri’s (2019) review of 55 empirical studies provided evidence to suggest that CT skills can be learned though Scratch in relation to Brennan and Resnick’s framework when taking the progression of learning into account. For example, children in the youngest group’s conceptual learning (5 to 9 years), were
limited to being able to read and construct sequences and algorithms related to directions, whereas older children (9 to 12 years) focused on ideas such as loops, events and conditionals. There have been fewer research studies conducted which examine teachers’ knowledge for teaching computational thinking in the primary classroom. Mason and Rich’s (2019) synthesis of the literature found 21 studies split across pre-service and in-service teachers whose findings suggest the importance of training and active participation to improve teachers’ computing self-efficacy (defined as a person’s belief that they “will be successful in using the technology” (Holden and Rada, 2011, p. 347)), attitudes and knowledge. The majority of the studies were short-term and focused on interventions to develop teacher’s content knowledge rather than their pedagogical knowledge which suggests a gap within the literature for this research area.

In the next section I bring together the fields of computational thinking and mathematical thinking by considering the ScratchMaths 5E pedagogical framework to combine Cuoco, Goldenberg and Mark’s (1996) definition of mathematical thinking exemplified through the eight habits of mind from (section 2.2.1 and Brennan and Resnick’s (2012) three key dimension computational thinking framework from section 2.3).

2.3.6 The intersection of computational thinking and mathematical thinking

The British Educational Communications and Technology Agency (BECTA), a non-departmental public body funded by successive UK governments between 1998 and 2011 had a responsibility to promote and to integrate ICT into the school classroom. BECTA’s (2003) report pointed to the clear benefits for educational technology use and cited research where technology that supported extended collaboration between pupils, aided communication of ideas and provided pupil motivation. Later publications led to providing guidance documents on a pupils entitlement to use ICT in primary mathematics (BECTA, 2009). With respect to mathematical thinking and computational thinking, the report is less clear aside from noting the development of visual imagery using Turtle Geometry. Hoyles and Noss (2003) approach to critiquing similar literature takes a more nuanced stance, focussing less on the effects of programming but drawing attention to what programming might bring to mathematics
education. Chief of these is the notion that by learning to program, children may develop a *Mathematical Way of Thinking* which can serve as a foundation for learning traditional mathematical content such as learning geometry (Papert, 1972). Papert continued to suggest, rather daringly noted Noss (1986) that perhaps instead of asking the traditional question of what are the transferable taught skills, whether it is possible to *use* algebra itself embodied in a computer programming language “as a vehicle for deliberately teaching transferable general concepts and skills.” (Papert, 1972, p. 251). This is certainly a proposition shared by Sherin (2001) whose own exploration of programming with Boxer (a new programming language building upon Logo where computational objects are represented in terms of boxes on the computer screen containing text, graphics or other boxes) aimed to replace traditional algebraic notation when representing the fundamental laws within physics laws. Sherin suggested that programming could shift the nature of mathematics and physics since “the understanding associated with programming-physics might be fundamentally different than the understanding associated with algebra-physics.” (Sherin, 2001, p. 1). This notion of using a computer programming language to represent and explain mathematical ideas is a key principle which underpins the ScratchMaths curriculum explained in more detail in Chapter 4.

Blackwell (2002) recast programming for research purposes proposing that when a user says they are programming, we should not consider whether the activity is genuine programming, but instead should focus on an analysis of the user’s experience in order to develop a more generic understanding of that programming activity. For example, from a user’s perspective the critical factor is not whether the problem is complex, but rather the cognitive resources that the user devotes to solving the problem at hand. The most well-known attempt to systematize categories of programmers at work is the framework defined by Turkle and Papert (1992), which proposed two types of programmer, *bricoleurs* (tinkers) and planners. Taking a different approach from previous research which attempted to categorise programmers (e.g. novices and experts in Soloway and Ehrlich (1984)), their goal was to put forth a notion of epistemological pluralism (Turkle and Papert, 1992) and show that both groups of students could achieve at a high level whilst taking very different approaches. Their argument was that whilst conventionally the planning behaviour is seen as superior (at that time the Harvard programming course mandated one way only), planning and tinkering should not be seen as desired
behaviours, but rather diverse ways of approaching the same problem, the *process of creating it*, without necessarily achieving superior results. This approach parallels that of mathematical thinking for professional mathematicians and is a design principle built into the ScratchMaths curriculum.

The programming behaviour of tinkering is encouraged by Scratch’s visual programming environment since a user can click on a block to explore its function as well as to combine blocks together into a larger project. The environment makes it easy to run scripts and to easily break them apart and reconnect them in the programming or debugging process. Similarly, the mathematical habit of mind of tinkering encourages students to take apart ideas and to put them back together. Students can see what happens when something is left out or if the pieces are put together differently. For example, if students are exploring the effects of the multiplication and division, they should first wonder if the order of multiplication and division matters, before questioning what happens when addition and subtraction is introduced. If the tinkering habit can be considered at the heart of thinking like a mathematician (Cuoco, Goldenberg and Mark, 1996), then tinkering can be similarly considered at the heart of programming like a computer programmer.

**2.3.7 The ScratchMaths 5E pedagogic framework**

The ScratchMaths *5Es framework* was developed for the ScratchMaths project when it became clear in the design year that teachers needed further guidance on effective pedagogical strategies in order to successfully implement the ScratchMaths curriculum (Benton *et al.*, 2016) and highlighted the importance of developing teacher knowledge. The constructionist theory (Papert, 1980) that underpins the curriculum, whereby pupils engage with mathematical ideas by building programs to explore them, is operationalised by use of the 5Es framework. The 5Es, consists of five unordered constructs, *Explore, Explain, Envisage, Exchange, and bridge*, based on research into good practice in teaching mathematics (see Benton *et al.* (2016); Benton *et al.* (2017) for further background). The 5Es are written into the learning objectives of each curriculum activity, the step-by-step activity instructions, as well as in the connections to mathematics sections of the teacher materials. The 5Es are defined as:
Explore – the importance of trying things out and using the computer to explore the learner’s thinking processes. Papert (1980) described the primary learning experience of Logo is about getting to know the Turtle and exploring what the Turtle can and cannot do. In the context of ScratchMaths, this constructionist approach encourages learners to first explore or to tinker with the actions and outputs of a single block to understand its function before moving to more complex programs where blocks are joined together. Blackwell (2002) describes the direct manipulation approach of a single action with a single visible effect as an important aspect of learning to computer program.

Explain – the importance of being able to explain what has been learned and articulating reasons behind a chosen approach. The teacher is encouraged to incorporate reflective questions and opportunities for discussion with peers or whole-class interactions, some of which are already built into the materials. Several researchers have highlighted the cognitive benefit of generating verbal explanations (Harel and Papert, 1990). In relation to mathematics Hoyles (1985) discussed how language can facilitate reflection and internal regulation, and how part of this process is the identification of which parts of the mathematical idea are important and which are not. Thinking about one’s own thinking is a key component of the constructionist approach (Han and Bhattacharya, 2001), as the programming language represents an object to think with. Explaining programs step by step is a key pedagogical approach to encourage the learner to think more deeply about the programming language beyond reading the label on the blocks.

Envisage – the importance of predicting an outcome before building or running the program and then compare the output to the prediction. In Scratch it can be very easy to snap blocks together and create attractive outputs without much knowledge of the underlying mechanism. This is an observation from my own teaching through programming and also supported by research into children’s understanding of a visual programming environment (Rader, Brand and Lewis, 1997). Playing Turtle, where the learner acts out commands given to the turtle, or uses physical objects such as a paper cut-outs, promotes body-syntonicity (Papert, 1980; Watt, 1998), whereby experiences are related to one’s knowledge and sense about one’s body. For ScratchMaths this would be knowledge of geometry when playing beetle.
2.3 Teaching and learning mathematics through programming

Exchange – drawing upon collaboration and sharing as a powerful way to learn, both as a constructionist practice advocating the development of ideas through interactions with others (Han and Bhattacharya, 2001) and working in cooperative small groups to support mathematical leaning (Swan, 2006). Within the ScratchMaths curriculum pupils are encouraged to work and discuss their strategies and discoveries in pairs, as well as for the teacher to display an example of a pupil’s program on the screen to discuss and potentially debug in a whole-class discussion.

Bridge – recontextualising the ideas that have been represented within the microworld using Scratch programming blocks into school mathematics. Powerful ideas should be embedded in any well-designed constructionist activity (Bers et al., 2014) and be seen as powerful through their connections with other disciplines such as mathematics (Papert, 2000). The SM teacher support materials makes explicit the mathematical links (see Chapter 4). Within the ScratchMaths curriculum, this could be an unplugged pencil and paper task away from the computer, or the teacher explicitly making the connection from programming to mathematics. The motivation for including bridge was that previous research rarely looked at making connections between mathematics and computing contexts.

Table 2.1 suggests a mapping of i) the ScratchMaths 5Es pedagogical framework, ii) Cuoco, Goldenberg and Mark’s mathematical habits of mind and and iii) Brennan and Resnick’s 3D computational thinking framework.

<table>
<thead>
<tr>
<th>Element of 5Es Pedagogical Framework</th>
<th>Element of Computational Thinking – CT Framework</th>
<th>Element of Mathematical Thinking - Mathematical habits of mind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explain</td>
<td>Perspective - questioning</td>
<td>Students should be describers Students should be pattern sniffers</td>
</tr>
<tr>
<td>Explore</td>
<td>Practice – being incremental and iterative Practice – abstracting and modularising</td>
<td>Students should be tinkerers Students should be inventors Students should be experimenters</td>
</tr>
<tr>
<td>Envisage</td>
<td>Practice – testing and debugging</td>
<td>Students should be visualisers Students should be conjecturers Students should be guessers</td>
</tr>
<tr>
<td>Exchange</td>
<td>Practice – reusing and remixing Perspective - expressing Perspective - connecting</td>
<td>Students should be describers</td>
</tr>
</tbody>
</table>
The mapping brings together the separate domains of computational thinking and mathematical thinking through the use of the 5E pedagogical framework which underpins the ScratchMaths curriculum materials used within this study. Notably, Pei, Weintrop and Wilensky (2018) took a similar approach to combine the mathematical habits of mind and computational thinking practices in their design of a microworld to explore geometrical concepts. Their findings suggested that their microworld LatticeLand provided a restructuration of geometry and showed how their new representational approach facilitated learners to develop computational thinking and mathematical habits of mind. The authors used Weintrop et al.’s (2016) framework (developed in parallel to that of the ScratchMaths project) which is not dissimilar to the above mapping, as the four main categories used: data practices, modelling and simulation practices, computational problem solving practices, and systems thinking practices are comparable to Resnick’s approach to CT. However, the framework was developed in the secondary context whereas the 5E pedagogical framework was developed for use in the ScratchMaths primary setting.

Commenting on the mapping in Table 2.1, the mathematical habits of mind and the 5E pedagogic framework are independent of the computing or mathematical concept being taught. This approach contrasts to Brennan and Resnick’s CT framework which separates the computational thinking concepts of sequences, loops, parallelism, events, conditionals, operators, and data as a separate dimension. It is therefore not appropriate to try to map this aspect of the CT framework since the 5Es are content independent. Similarly, the ScratchMaths bridgE pedagogical approach is unmapped since bridge provides a unique pedagogical approach to bridge between computing and of mathematics contexts in both directions and therefore cannot exist in only context. It is clear from Table 2.1 that the mapping is not necessarily a 1-1 mapping and that some mappings may be more strongly connected than others. For example, the ScratchMaths pedagogical practice of Exchange, encourages pupils to share their computer programs with each other. This practice is equivalent to the CT practice of re-use/remixing code created by others. Within the 5E framework this can be seen as a social
exchange such is as defined by the mathematical habit of mind of describing which encourages collaboration with others. However, the expressing perspective of CT focuses on seeing technology as something that can be used for design and self-expressing, which is perhaps less strongly connected to simply collaborating or sharing with others. Despite possible limitations of the mapping, what the mapping provides is a theoretical position to bring together the fields of mathematical thinking, computational thinking in the context of the ScratchMaths curriculum and contributes to the development of the conceptual framework underpinning this study discussed in section 2.4.

2.4 Towards a conceptual framework for analysing teachers’ knowledge for teaching mathematics through programming in the primary setting

The importance of the role of the teacher as a theme within mathematics education research has grown in significance over the last thirty years. The research has shown a link between teachers’ knowledge, their classroom practice and student achievement (Hill, Rowan and Ball, 2005; Hodgen, 2011; Mason, 2008; Petrou and Goulding, 2011; Rowland and Ruthven, 2011). The ScratchMaths curriculum intervention was evaluated in 2018 by the Education Endowment Foundation (EEF), who tested whether ScratchMaths could be used to improve pupils’ computational thinking skills, and whether this in turn could have a positive impact on Key Stage 2 maths attainment. Given that the independent evaluation focused only on the attainment of the children, and ignored any aspect of teacher practice, the project provided a unique opportunity to research the critical role of the teacher, and in particular study the classroom practices of teachers via their evolving teacher knowledge as they engaged with ScatchMaths to reach mathematics through programming.
2.4.1 Mathematical knowledge for teaching (MKT) framework

There have been various attempts to categorise and map out teacher’s professional knowledge. The most influential is that of Shulman (1986) who organised professional knowledge into three principal domains: Subject Matter Knowledge (SMK), Pedagogical Content Knowledge (PCK) and Curricular Knowledge (CK). He raised the criticality of PCK as a type of knowledge that went beyond knowing the subject and was required in order to teach the subject. However, the classification did not define any detail of the specifics of the knowledge required to teach and did not specify if the subject discipline affected the knowledge required. To address these limitations mathematics education researchers (Ball, Thames and Phelps, 2008; Baumert and Kunter, 2013; Davis and Simmt, 2006; Ma, 2010; Rowland et al., 2009; Schoenfeld and Kilpatrick, 2008) have developed the ideas of PCK into the domain of mathematics teachers’ knowledge for teaching. Their frameworks differ by contextualising the knowledge for primary and secondary mathematics classrooms as well as foregrounding different aspects of the knowledge required. For example Mathematical Knowledge for Teaching (Ball, Thames and Phelps, 2008) considers SMK and PCK but breaks down SMK into various subdomains of mathematical knowledge whilst Rowland et al’s (2009) Knowledge Quartet identifies situations in which such knowledge can be seen in the act of teaching.

The mathematical knowledge for teaching framework (MKT) (Ball, Hill and Bass, 2005; Hill and Ball, 2004) developed Shulman’s categories and clarified the distinctions between SMK and PCK. Important to the framework was the development of a set of measurement instruments for the assessment of elementary school teachers’ MKT (Ball and Bass, 2003; Hill, Schilling and Ball, 2004; Hill et al., 2007). Their model suggests that SMK can be divided into Common Content Knowledge (the mathematical everyday knowledge that all educated adults should have), Specialized Content Knowledge (the specialist knowledge acquired through professional training and classroom experience, and Horizon Knowledge, defined as sequencing of the curriculum. In addition, they placed an emphasis on knowledge of the students by dividing PCK into three further categories: Knowledge of Content and Students, Knowledge of Content and Teaching, and Knowledge of Content and Curriculum (Ball, Thames and Phelps, 2008). Research by Baumert et al. (2010) supported the MKT
categorisations and found that teachers’ pedagogical content knowledge was theoretically and empirically distinguishable from their content knowledge.

Hill, Rowan and Ball’s (2005) study of elementary school teachers examined the predictive validity of mathematical knowledge for teaching and student’s learning gains in mathematics in a large-scale study of 115 elementary schools. The team developed a series of multiple choice items which could be used to measure mathematical knowledge for teaching such as questions which were purely content based, e.g., “What power of 10 equals 1?” and those that were more situational, such as responding to different student methods for written multiplication. Hill, Rowan and Ball’s multilevel analyses of the data showed that elementary teachers MKT positively predicted students’ learning gains in two different grades of school. Conversely, students who were taught by teachers who scored at the lower end of the MKT distribution made considerably fewer gains in the standardised testing scores than those teachers at the higher end of the MKT distribution. Their research also found a positive effect of the role teachers’ content knowledge played in the teaching of elementary mathematics content in the early grades of school. This finding is particularly important to this study since the mathematical focus of this research is teachers’ mathematical knowledge for teaching place value and the early algebraic concepts of variable. The MKT framework is based on teaching mathematics without the use of technology; teaching using technology in the mathematics classroom therefore requires another dimension of teacher knowledge to be explored.

2.4.2 Teacher knowledge for teaching mathematics with technology frameworks

The successful integration of digital technologies into everyday teaching practice relies crucially on the relevant experience on the part of the teacher (Ruthven, 2014). Ruthven has outlined three contemporary frameworks for analysing teacher’s expertise, the Technology, Pedagogy and Content Knowledge (TPACK) framework (Koehler and Mishra, 2009), the Instrumental Orchestration framework (Trouche, 2005) and the Structuring Features of Classroom Practice framework (Ruthven, 2009). This section will provide a short discussion
of each of the frameworks and how they informed the conceptualisation of teacher knowledge used within this thesis.

Historically, technology tended to be a separate domain of knowledge to be learned, unintegrated into the domains of content and knowledge of pedagogy. As Shulman had initially defined PCK to distinguish the how to teach rather than what of teaching, researchers similarly focused on the integration of technology, content and pedagogy (Koehler and Mishra, 2005; Niess, 2005; Pierson, 2001). An initial model that conceptualised the body of knowledge teachers need for teaching with and about technology was an amalgam of three knowledge elements: technological pedagogical content knowledge (TPCK). TPCK is most commonly represented as a Venn diagram with three overlapping circles, each representing a domain of teacher knowledge. Mishra and Koehler (2006) developed the conceptualisation to include the interactions of the three core categories which resulted in four additional types of knowledge: pedagogical content knowledge, technological pedagogical knowledge, technological content knowledge, and technology pedagogical content knowledge.

The TPCK framework, or more recently in the literature as Technology, Pedagogy and Content Knowledge (TPACK), draws attention to the “Total PACKage” (Thompson and Mishra, 2007) of knowledge and is a useful heuristic to raise questions about the interaction of the various knowledge elements. However, researchers have raised issues of the coarse grain nature of the framework to conceptualise and analyse teacher knowledge (Graham, 2011; Ruthven, 2014). An important limitation of TPACK for the context of this study is that the knowledge is considered generic across all subject domains and does not capture any of the subtle nuance of mathematical knowledge. There are a number of mathematics specific variants such as Niess at al.’s (2009) Mathematics TPACK which introduced descriptor levels or Landry’s (2010) mathematics focused TPACK teacher survey. However, both models, despite the mathematics focus share a limitation in that they do not consider the personal beliefs or orientations of the teacher themselves.

Research from Zbiek and Hollebrands (2008) identified how the teachers’ perception of the nature of the mathematical knowledge and how it should be learned, impacted the teachers’ decision to use technology in the classroom. A consideration of the factors influencing teacher
Towards a conceptual framework for analysing teachers’ knowledge for teaching mathematics through programming in the primary setting

Use of technology led Thomas (Hong and Thomas, 2006; Thomas and Hong, 2005) to propose a framework for mathematical pedagogical technology knowledge (PTK) as shown in Figure 2.1 below.

![Diagram of PTK framework]

Figure 2.1 A model of the framework for PTK (Thomas and Palmer, 2014, p. 76)

Thomas and Palmer (2014) defined a teacher’s PTK as incorporating “the principles, conventions and techniques required to teach mathematics through the technology.” (Ibid, p. 75). Therefore, the model not only includes the need to be proficient user of technology but also the need to understand the principles and techniques required to create situations which enable mathematical learning through technology. The PTK framework places greater emphasis on teachers’ mathematical and pedagogical knowledge through the incorporation of Ball and Bass’ Mathematical Knowledge for Teaching framework. The PTK model also employs the theoretical basis of the Instrumental Approach, in which the digital tool and person co-evolve so that what starts as a crude instrument is transformed into a functional instrument through a process of Instrumental Genesis (Artigue, 2002; Guin and Trouche, 1998). Whereas the TPACK models are related to the knowledge and existence of components and capabilities of technology and how they might change as a result of using particular technologies (Mishra and Koehler, 2006), PTK puts a greater emphasis on the emphasis on the epistemic value of
the technology to produce knowledge of the mathematical object under study (Artigue, 2002; Lagrange et al., 2003).

The instrumental approach as applied to mathematics education was based on seminal studies of French upper secondary mathematics students as they used graphics calculators to study functions. Since that time the focus has moved to include the role of the teacher. The concept of instrumental orchestration (Trouche, 2004) emerged to put emphasis on the necessity of the teacher in steering their students’ instrumental genesis. An “instrumental orchestration is defined as the teacher’s intentional and systematic organisation and use of the various artefacts available in a computerized learning environment in a given mathematical task situation, in order to guide student’s instrumental genesis.” (Drijvers et al., 2009, p. 2). An instrumental orchestration consists of two components a didactical configuration and an exploitation mode (Trouche, 2004) A didactical configuration is the arrangement of artefacts in the environment which includes the technological tools buts also the tasks students work on. An exploitation mode is the way the teacher decides to exploit the didactical configuration for their didactical intentions.

The theory of instrumental orchestration was used to investigate the types of orchestrations teachers developed when using technology to teach mathematics. Drijvers’ research led to the identification of eight whole-class orchestration types, Technical-demo, Explain-the-screen, Link-screen-board, Discuss-the-screen, Spot-and-show, Sherpa-at-work, Guide-and-explain and Board-instruction (Drijvers et al., 2009; Drijvers et al., 2010; Drijvers et al., 2013). For each of the whole-class orchestration types, the didactical configuration includes the teacher’s access to technology to project the computer screen, and a classroom arrangement that allows students to follow what is on the screen. The exploitation mode includes the use of teacher tasks or student work. The whole-class orchestration types:

- **Technical-demo** concerns the demonstration of tool techniques by the teacher.
- **Discuss-the-screen** concerns a whole-class discussion of what is on the computer screen.
• *Explain-the-screen* concerns a whole-class explanation by the teacher guided by what is on the computer screen. The explanation involves mathematical content beyond technical techniques.

• *Guide-and-explain* holds the middle ground between Explain-the-screen and Discuss-the-screen by providing a somewhat closed explanation of what is on the screen rather than an open discussion. There is therefore limited student-teacher interaction.

• *Link-screen-board*, the teacher stresses the relationship between what happens in the technological environment and how this might be represented in the conventional mathematics of pencil and paper on the board.

• *Spot-and-show*, the teacher uses student work to deliberately bring students reasoning to the fore through classroom discussion.

• *Sherpa-at-work*, the teacher asks a so-called Sherpa student to operate the technology to either present their work or carry out the teacher requested actions.

• *Board-instruction*, the teacher makes no connections to the use of technology and uses the board for writing as in how they might regularly teach the whole-class.

Drijvers et al. (2013) did not claim completeness for the inventory of whole-class activity categorizations. For example, Tabach (2011) added the important *Not-use-tech* orchestration to describe a situation when a teacher chooses not to use the technology despite the technology being present within the classroom. However, the instrumental orchestration framework provides an important theoretical approach to analyse teachers’ pedagogical practices in the context of teaching mathematics using technology. Since Drijvers et al.’s typology was developed in the analysis of lower secondary school teachers use of Java applets to explore mathematical concepts of function, this research provides an opportunity to apply (and possibly develop) the whole-class instrumental orchestrations for teaching mathematics through programming in the primary setting.

Ruthven (2009) developed a third framework designed to support the identification and analysis of the teaching expertise required for integration of new technology, the Structuring Features of Classroom Practice framework. The framework identifies “five structuring features
of classroom practice which shape the ways in which teachers integrate (or fall short of integrating) new technologies: working environment, resource system, activity structure, curriculum script and time economy.” (Ruthven, 2014, p. 386). Given that the ScratchMaths intervention did not require the teachers to choose or devise their own tasks (a key characteristic of the structuring feature “curriculum script”), all had to provide classrooms which would give pairs of children to working on a single computer and provide appropriate curriculum time (all characteristics of the structuring feature “working environment”), I decided that this framework would not be appropriate for analysing teaching expertise.

### 2.4.3 Mathematical Pedagogical Technology Knowledge (MPTK) framework for teaching mathematics through programming in the primary setting

I have selected Thomas and Palmer’s PTK framework as the main theoretical basis for my conceptual framework to examine teachers’ evolution of knowledge as they engage with teaching mathematics through computer programming. The framework has been selected as it foregrounds teachers’ mathematical knowledge as well as the notion of how the use of technology shapes the mathematical knowledge required for teaching through programming; both of these aspects are key to this study. Clark-Wilson and Hoyles (2017) used the PTK framework to examine teachers’ knowledge and practices when teaching with dynamic mathematical technology which offered manipulable mathematical representations. Their findings indicated the addition of a bi-directional connection between Mathematical Knowledge for Teaching and Technology instrumental genesis to capture the “process through which a teachers’ existing MKT impacts on the knowledge that they bring to their early technology experiences” (Clark-Wilson and Hoyles, 2017, p. 85). This is an important addition to the model since the focus of my research is the teachers’ evolution of knowledge which may be shaped and shaped by their use of computer programming to teach mathematics. They added (Mathematical) in parenthesis, hereafter known as the Mathematical Pedagogical Technology Knowledge (MPTK) framework.

Since my study is interested in analysing mathematical and computing knowledge in the context of programming it is necessary to modify the MPTK framework and include the
additional knowledge component *Computational Content Knowledge* which considers the subject knowledge required to teach computer programming (Figure 2.2). The figure also shows how the topics of variable, place value, angle and mathematical and computational thinking are related to each other by linking to the corresponding knowledge domains. For example, the concept of variable is connected to the mathematical content knowledge and to the computational content knowledge. The intersection of mathematical and computational thinking as discussed in section 2.3.6 is represented by a bi-directional arrow between the two components. The *Mathematical/Computational Knowledge for Teaching* component of the model brings together both the mathematics and computational knowledge with the corresponding pedagogical knowledge. A finer grained *a priori-view* of how I analysed the teachers’ knowledge for each of the components is discussed in Chapter 4 - Methodology.
Figure 2.2 Mathematical Pedagogical Technology Knowledge (MPTK) framework for teaching mathematics through programming in the primary setting

2.5 Research questions

In summary, the literature review has positioned the block-based programming language Scratch as a tool to research and to build upon the significant prior research of using Logo programming to teach mathematics. The literature has also identified the critical role of the teachers’ knowledge and supported the development of a conceptual framework that can be
utilized to analyse teachers’ knowledge as they engage with teaching mathematics through programming.

The key ideas from the literature which underpin this study:

- The continued importance of developing mathematical thinking and reasoning;
- The critical role of the teacher to make connections within mathematics;
- The role of representations to support mathematical understanding;
- The conceptual mathematical knowledge of the place value system is well understood;
- Teachers’ own knowledge of place value can limit their own capacity to make connections and to link representations;
- Existing approaches to teaching place value may limit teachers’ capacity to confront and reconceptualise their existing knowledge;
- Variable is a challenging area of mathematics and computing; multiple misconceptions are held in both contexts;
- Conceptualizations of angle can be described as i) a dynamic notion of an angle as movement ii) a measure such as the length of arcs and iii) a (static) geometric shape such as delineation of space between intersecting lines;
- Developing mature abstract angle concepts depend essentially on learning to link together the various angle perspectives in different angle contexts;
- The use of constructionist practices through computer programming to develop computational and mathematical ideas;
- The critical role of the teacher and of the teachers’ knowledge to teach effectively with technology;
- The limited studies which research the development of teachers’ knowledge in the primary setting as they engage with learning to teach programming to explore mathematical concepts;

As discussed within the personal motivations for the study of the introduction, I was part of the ScratchMaths team who designed a two-year computing and mathematics-based curriculum
intervention which utilised the Scratch programming language to develop computational thinking skills and to explore mathematics concepts. I also developed and provided national professional development for the teachers who would teach the intervention. The project’s primary research output focused upon the children, in particular the evaluation of the intervention which considered the children’s performance in the end of Key Stage 2 in their mathematics assessment. However, this research focuses on the teacher and their evolving mathematical pedagogical technology knowledge as they engaged with the ScratchMaths curriculum materials to teach activities on place value, variable and angle. The key ideas in the literature have led to the development of a conceptual framework (see Figure 2.2 above) and gave rise to two research questions:

- (RQ1) How does primary teachers’ knowledge and mathematical thinking of selected mathematical concepts shape their engagement with teaching microworlds around these concepts, and reciprocally, how is their knowledge and thinking shaped by this engagement?
- (RQ2) How do primary teachers use the digital resources of the ScratchMaths curriculum in their teaching to orchestrate children’s learning?

Note: at the time of writing the literature review there was only one research study (Choi, Jung and Baek, 2013), which specifically reported upon designing and implementing an intervention for learning mathematics with Scratch. The preliminary study lacked a conceptual framework or theoretical underpinning which led the authors to conclude that the findings should be treated with caution. However, they did suggest that an instructional approach of introducing the mathematics, followed by building a model in Scratch could be used to develop problem solving, logical thinking and develop a positive attitude towards mathematics. The few research studies that have emerged using Scratch programming to explore mathematics since this literature review will be addressed in the conclusion chapter 10.
2.5 Research questions

...
Chapter 3  Research methodology and methods

3.1  Introduction

In this chapter, I describe the research methodology I adopted to answer my research questions. I begin the chapter by outlining and justifying the research paradigm, the research design, and the methods selected for data collection. Next, I introduce how I operationalised my conceptual framework for teaching mathematics through programming in the primary school setting discussed in section 2.4. Finally, I explain my process for data analysis as well as provide a consideration of ethical issues.

3.2  Research paradigm and design of the study

This research adopts a design-based implementation research (DBIR) paradigm. The DBIR approach focuses on working with practitioners to develop and test designs for improving teaching and learning. Penuel et al. (2011) articulated the four principles that underlie DBIR i) a focus on persistent problems of practice, as experienced from multiple stakeholders’ perspectives, ii) a commitment to iterative, collaborative design in realistic contexts, iii) a concern with developing theory and knowledge related to both classroom learning and implementation and iv) a concern with developing capacity for sustaining change in educational systems. The rationale for the ScratchMaths project, driven in part by the 2012 UK government policy decision to introduce a new computing based curriculum is underpinned by principles i) and iv). Principle ii) underpins the design and development of the ScratchMaths curriculum materials, developed and tested iteratively with teachers from ‘design schools’ (see Benton et al. (2017) for a more detailed discussion of the DBIR approach to design and testing of the materials). The third principle’s focus on developing theory and knowledge related to
classroom learning and implementation and the second’s focus on iterative design in realistic contexts is particularly pertinent to my study.

The participant teachers of this research are also participants in the ScratchMaths intervention and are positioned as ‘designers of lessons’ who will (re)conceptualise the SM curriculum materials according to the constraints of their unique classroom settings (Laurillard, 2013). The teachers’ adaptation and implementation of the ScratchMaths modules provides an insight into their classroom practices as they foreground mathematical or computational content and pedagogical approaches that they perceive to be important. Given that using Scratch for both the purposes of teaching computing and teaching mathematics is a new innovation, it is inevitable that some elements of the teacher’s established practices will be in tension with the ScratchMaths approach. This was an issue also faced by the Cornerstone Maths programme (Clark-Wilson et al., 2015). Consequently, developing a research design which opens a window into teachers’ mathematics pedagogical technology knowledge (MPTK) will enable me to trace the evolution of teachers’ mathematical knowledge as they learn to program using Scratch.

Cohen, Manion and Morrison (2018, p. 173) remark that there “is no single blueprint for planning research. Research design is governed by ‘fitness for purpose’”. Therefore, the purposes of this research will determine the design of the research and inform the methodology, which in turn informs the methods and data collection instruments. My doctoral research aims to trace the knowledge and classroom teaching practices of primary school teachers as they engage with the carefully sequenced ScratchMaths activities to explore mathematical ideas through computer programming. More specifically, I seek to provide an in depth exploration of teachers’ MPTK as they teach activities from modules 4 and 5 of the ScratchMaths curriculum to engage with mathematical thinking and reasoning around the mathematical concepts of place value, variable and angle. Given that my research aims to investigate a contemporary phenomenon in its real-world context (Yin, 2014), i.e. teachers in their natural classroom setting teaching computer programming to explore mathematical concepts, and will rely on myself as the researcher as a key instrument to collect data using multiple methods (Creswell, 2018), I chose a qualitative inquiry with a multiple-case study design.
Yin (2018) argues that although there is no formula for deciding to use case study research, the research questions themselves can lead to the decision to use case study research. He explains that the more the questions seek to explain a contemporary phenomenon such as how something works, the more that case study research will be relevant. My overarching objective is to identify what mathematical, and computing teaching knowledge is required to teach ScratchMaths activities on place value, variable and angle effectively. Although this objective might be considered a “what” question, and consequently lead to a survey method, my use of “what” is in an exploratory sense and is such a justifiable rationale for conducting an exploratory case study. My two research questions seek to explore how teachers’ MPTK around mathematical concepts evolves and how teachers use programming as they teach with digital activities from the ScratchMaths curriculum. The case study approach thus allows a researcher to portray, analyse and interpret real individuals through accessible accounts to capture the complexity and situatedness of behaviour, in contexts that are unique and dynamic (Cohen, Manion and Morrison, 2018).

Qualitative case study approaches must include ‘thick descriptions’ (Geertz, 1973) of the contextualised teacher behaviour. Thick descriptions should include detailed observational data as well as data on meanings and participant’s interpretations of situations and potential unobserved factors. An important consideration is that the attribution of meaning is continuous and evolving over time (Cohen, Manion and Morrison, 2018). This consideration supported my decision for the case studies to be longitudinal in nature so as to trace the evolution of teachers’ MPTK as they engage with Scratch and the ScratchMaths curriculum materials. The rich and thick descriptions in each case will help to identify any differences or similarities between the cases. In addition to the rich cases, I have chosen to augment the cases studies with additional teacher episodes to fill any gaps that arose in the data and to provide additional exemplification of emerging ideas. This teacher episode group of teachers were observed and interviewed only once over the course of the study and would therefore provide an insight into teachers’ practice and their evolving MPTK where there had been no researcher intervention.

The qualitative case study approach necessitates integrating a variety of data collection methods in order to triangulate and corroborate data by seeking different data about the same
phenomenon or to answer different research questions (Mason, 2017; Yin, 2018). For my research, it was essential to use multiple methods for data collection to ensure that different perspectives of teachers’ knowledge were explored. For example, teacher knowledge that could be observed in the classroom setting as well as teacher knowledge demonstrated in an interview or task-based interview. Structured, task-based interviews involve the teacher interacting with the researcher in relation to one or more tasks introduced in a preplanned way (Goldin, 2000). The triangulation approach of collecting multiple sources of evidence contributes towards addressing issues of reliability of validity of my research. For each case study, the data collection consisted of audio or videoed lesson observations (dependent on the level of comfort of the observed teacher), post-lesson semi-structured interviews about the observed lessons and the related mathematical concepts, and task-based interviews of teachers working directly with ScratchMaths activities. See section 3.4 for the details of the data collection methods. I had intended to collect lesson planning materials from the teachers; however, all teachers only used the provided ScratchMaths curriculum materials, e.g., starter projects, power point student presentations, teacher guide, and did not create any additional written plans. In summary, by utilising different methods I was able to collect rich data of teachers teaching activities from the ScratchMaths curriculum in their real-life classroom contexts.

Yin (2014) makes a compelling argument with respect to the generalizations that can be drawn from a case study approach. He makes an important distinction between analytic generalisation and statistical generalization. The latter form of generalization is that which is typical of empirical research, e.g., an inference is made about a population based on a sample of evidence collected from that population. His argument is that this would be a fatal flaw of doing case studies since the sampling unit would be too small to adequately serve as a representation of a larger population when using statistical generalisation. However, rather than thinking of the case as a sample, he explains that it should be thought of as the opportunity to “shed empirical light about some theoretical concepts or principles, not unlike the motive of a laboratory investigator … in conducting a new experiment” (Yin, 2014, p. 40). Analytic generalisation is based on corroborating, modifying, rejecting or advancing theoretical concepts used in the design of the case study, or based on outlining new tentative theory, identifying mechanism that explain the phenomenon that arises from completing the case study. A multiple-case study
approach with the augmenting teacher episodes therefore strives for analytic generalisations to be at a conceptual level higher than that of the specific case or subject participant to support theory elaboration.

In sections 2.4 I discussed the strengths and limitations of existing models of teacher knowledge for teaching mathematics with technology since each model could not be applied to the specific case of teaching mathematics through computer programming without adaptation. Consequently, I proposed a conceptual framework for teaching mathematics and computing in the primary school context which underpins this thesis. My research necessitates the use of a multiple-case study with augmenting teacher episodes to test the conceptual framework by utilising analytic generalisation to compare and contrast differences, along with similarities among the multiple cases and teacher episodes (Ridder, 2017). Additionally, the multiple-case studies and augmenting teacher episodes will enable me to illustrate how the theoretical aspects of the model, such as teachers use of instrumental orchestrations, or aspects of teacher knowledge at the intersection of mathematics and computing apply in different classroom settings.

### 3.3 Selecting participants for the study

I adopted a purposive sampling approach to focus on a specific unique case, and to generate theory through the accumulation of data from different sources (Teddlie and Yu, 2007). Within the English school system, children aged nine and ten are in Year 5 of primary school and children aged ten and eleven are in Year 6, the final year of primary school. The specific unique case for my study would be participants selected from the ScratchMaths (SM) intervention who continued to teach the SM curriculum into Year 6. The modules in Year 6 had been specifically designed to explore mathematical ideas explicitly and to build upon the Year 5 modules where computational thinking was foregrounded, but mathematical ideas were implicit. Given the mathematics focus of this study selecting the unique case from this population was appropriate. I chose characteristics of the population to ensure that the teachers would have a high degree of fidelity of the SM intervention, i.e., those teachers that had been teaching SM for at least a year and had attended most of the professional development. Thus, the teachers would be in a
similar position to their children: i) they would have knowledge of the SM curriculum from Year 5, ii) they would have developed confidence with Scratch, iii) they would have developed computational skills to build upon and express mathematics within Year 6. Criteria for school inclusion in the SM intervention required schools to be a two form entry and as a result expected that two teachers per school would be involved in the intervention to support collaborative working. Consequently, I intended to have both teachers from the school be part of the sample.

The population of teachers to sample from would therefore satisfy four characteristics:

- attended at least 2 of the 3 SM professional development opportunities during Year 5,
- taught at least one module from Year 5,
- attended both of the SM professional development opportunities before commencing Year 6,
- planning to teach at least one module from Year 6.

From the 50 treatment schools of the SM two-year intervention (another 50 schools formed the control group), there were 10 schools which satisfied the four characteristics. I decided that 8 schools and the corresponding 16 teachers would be selected as this would be a manageable number for a sole researcher working in full time employment and would provide manageable national travel. Eight schools were then randomly selected. However, not every school had assigned two teachers to teach SM. The reasons for this were due to staffing decisions made in the school, e.g., maths was taught to both classes by the teacher considered to be the best or most experienced maths teacher in part due to ensuring that children were as best prepared as they could be for the KS2 maths examination. In just under half of the schools one teacher was assigned to teach SM to both year 6 classes. Although this was not ideal since my initial intention was to contrast teachers within the same school, I felt that there would be additional contrasting opportunities since a teacher may teach the same lesson during one observational visit.

Each teacher was observed during the Spring Term and a second observation planned for later in the same term. However, many of the second observations were postponed by the teachers due to reasons often beyond their control such as school trips organised on the same as their SM lessons (often a Friday), or other school activities which took precedence. The observations
that had been postponed to the Summer term were frequently cancelled due to the impact of preparation for the KS2 tests and SM lessons cancelled. However, two schools in North London and their three teachers, Mary, Rina and Sally re-scheduled their lessons and allowed me to continue to observe and interview them across the school year. As a result, data collected from the three teachers formed the three main cases of this study (Chapter 5, 6, 7), and the additional data from the other teachers augment the cases as a series of teacher episodes used to plug any gaps in the data and to colour the landscape and provide additional exemplification of ideas (Chapter 8).

### 3.3.1 The case study teachers

In this section I present the backgrounds and profiles of the three case study teachers. They are anonymised by using pseudonyms: Case study 1: Mary; Case study 2: Rina; Case study 3: Sally.

**Case study 1: Mary**

Mary was a Year 6 teacher in her 9th year of teaching and worked at Diamond Primary school in London. Diamond Primary is a larger than average three form entry school of 360 children. She was aged between 25-35 years old and held an undergraduate degree in Public Administration and a primary PGCE. Mary’s experience with Scratch programming prior to her involvement in the SM project had been limited. In an early interview she explained that when the national computing curriculum requirements changed, she was appointed as the school’s computing coordinator and given minimal professional development. Mary explained:

> I did a Scratch INSET but my knowledge was very limited. I could pick it up by following the plans, but a lot of that would have been looking, and seeing, it wouldn’t be the understanding I have now [from ScratchMaths] of why I’m doing things. It was more, this is what the teachers book says, I’ll do this. [DMW4(Int)]

Although Mary had attended all the Year 5 SM professional development, in the first year of the intervention, she did not teach the Year 5 curriculum due to being staffed as a Year 6 teacher. However, Mary had supported the Year 5 partner teacher in all their lessons and had overall oversight of the delivery of the curriculum. I considered that this specific case was equivalent to having taught at least one of the Year 5 modules as set out in my purposive
criteria. Mary was notably quiet during the CPD, choosing to complete the activities and to listen rather than contribute by sharing her thoughts and ideas actively. She said that this is typical of her approach to courses, and that she would rather sit at the back and listen [DMW3 (Int)]. However, Mary was passionate about technology usage which led to the creation of her role as computing coordinator and the school pursuing the offer of being part of the trial. Her interest in learning more about programming and the lack of confidence to teach with technology of her partner Year 6 teachers led Mary to be timetabled to teach all three of the classes in Year 6.

During the data collection period, Mary was interviewed and observed teaching SM materials which used the place value and fireworks microworlds from Modules 4 and 5. Mary taught her own Year 6 class as well as two other classes from the same year group.

**Case study 2: Rina**

Rina was a Year 6 teacher in her late-twenties at Emerald Primary School in London. Emerald is a larger than average primary school, it has two forms of entry as well as a class of children in Nursery. Rina was in her seventh year of teaching after obtaining a BSc in Zoology and completing a PGCE in primary education. Rina studied mathematics to A level and stated that her mathematical competency was largely a result of the of statistics within her undergraduate science degree. Rina had some previous teaching experience of computing as she had run a children’s afterschool club to create robots using LegoNXT.

In 2013 when the new compulsory national curriculum for computing was introduced, her school had provided an INSET session on how to use Scratch. After the training, Rina had used Scratch in a science lesson to teach a mixed Year 5 and 6 class to create multiplying bacteria animations using sprites. She had some limited experience teaching a computing curriculum developed within her school which drew upon freely available resources.

Rina taught the three Year 5 SM modules in year one of the intervention and reported that she felt very confident in teaching SM. She played an active role in the CPD, sharing her Year 5 experiences and explained her response to the gap task (completed by 50% of attendees), detailed in Chapter 4. Rina explained that she had developed a no-nonsense approach to using
computers in the classroom. She explained that she learns how something works by pressing all the buttons to see what they do, so that when a child is stuck, she first encourages them to experiment. Rina explained:

“I’d be the kid that took the toy apart. I know I’m computer literate because I’ve broken so many computers by changing everything. So, when someone comes to me and says “I can’t do something”, I’m a bit like, ‘Well why haven’t you pressed all the buttons and worked out what they all do?’” [ERE2(Int)]

During the data collection period, Rina was observed teaching SM using the place value and fireworks microworlds from Modules 4 and 5. Rina taught the same Year 6 class who she had taught as Year 5 in the previous year.

Case study 3: Sally

Sally was a Year 6 teacher in her late-twenties also at Emerald Primary School in London. She had six years teaching experience and holds a BA in Education. Sally was appointed to Emerald school as a Year 6 teacher and as the computing lead to be responsible for the school computing curriculum and technology implementation. Sally’s academic background was primarily social sciences, studying psychology and health and social care. She had not studied mathematics beyond aged 16 and was unfamiliar with the mathematical content studied at that level. However, she hoped that by developing children’s resilience to failure and by developing children’s reasoning and problem-solving skills would enable them to cope with the demands of higher-level study. Sally explained that her interest in technology and computer programming was sparked during her degree at a demonstration of computer software which allowed the user to program on screen buttons to control lights and simple motors of other on-screen components. As was typical of the time before computing was compulsory in the curriculum, this was the extent of her initial teacher training in technology and computer programming. She had attended some continuing professional development sessions on the educational use of technology but considered that her programming skills to be self-taught.

Sally did not attend any of SM professional development since she joined the school after the main PD had been given. However, was expected to teach the Year 6 SM curriculum alongside her Year 6 partner teacher Rina. Although this meant that she did not meet the purposive criteria for inclusion within the sample, her partner teacher at the school did, and as such I felt that this
was a possible extreme or deviant case to research (Flick, 2018) and would provide further comparison opportunities with the other typical cases. For example, in the first term at the school, Sally had tried to teach using the place value microworld. Her year 6 class of children had been taught SM by a different teacher in the previous year and therefore had significantly more Scratch and SM experience than Sally. She reflected after the lesson that she felt very unprepared as she had not received any SM professional development or was aware of the supporting teacher materials. Sally explained:

I had no idea what I was doing… I wasn’t doing it how I should have been, and I had no idea where it was going. The kids had previous experience. Because they were doing something in the lesson, we just carried on! [ESE5(Int)].

During the data collection period, Sally was observed teaching SM materials using the fireworks microworld from Module 5.

The teacher episodes

The teachers who are the subject of the augmenting teacher episodes have been organised into two groups for discussion and analysis. Table 3.1 below provides a profile for each of the teachers within each group. The groups are based on their prior experience of SM and the associated professional development.

Group A teachers: little or no prior experience of SM:

- attended some to none of the Y6 professional development (PD),
- not taught the Year 5 SM curriculum.

Group B: some prior experience of SM:

- attended the majority of the SM Y5 and Y6 PD,
- taught the Year 5 SM curriculum or worked closely with the teachers that had.
### Table 3.1 Teachers, schools and profiles for Group A and Group B of the case study sketches

<table>
<thead>
<tr>
<th>Group</th>
<th>School</th>
<th>Teacher</th>
<th>Profile</th>
<th>Microworld</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>Humite</td>
<td>Betty King (BK)</td>
<td>&lt;5 years teaching experience. Arrived at school during Year 6. Missed Year 6 PD. No previous Scratch or programming experience.</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>Freshwater</td>
<td>Anthony Oliver (AO)</td>
<td>5-10 years teaching experience. Attended Year 6 PD. No previous Scratch or programming experience.</td>
<td>Place value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Priya Rashmi (PR)</td>
<td>&lt;5 years teaching experience. Attended Year 6 PD. No previous Scratch or programming experience.</td>
<td>Place value</td>
</tr>
<tr>
<td></td>
<td>Garnet</td>
<td>Elizabeth Angel (EA)</td>
<td>&lt;5 years teaching experience. Missed Year 6 professional development. Limited programming experience other than an introductory session during teacher training.</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trevor Trafford (TT)</td>
<td>HTLA covering long term sickness of staff member. Missed Year 6 PD. Some self-taught programming knowledge.</td>
<td>Variable</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>Carnelian</td>
<td>Katherine Williams (KW)</td>
<td>&lt;5 years teaching experience. Taught Year 5 ScratchMaths, attended majority of Year 5 and Year 6 PD.</td>
<td>Place value</td>
</tr>
<tr>
<td></td>
<td>Amber</td>
<td>Laura Havers (LH)</td>
<td>&lt;5 years teaching experience. Taught Year 5 ScratchMaths to two classes, attended majority of Year 5 and Year 6 PD. Sole ScratchMaths teacher in school.</td>
<td>Place value</td>
</tr>
<tr>
<td></td>
<td>Iolite</td>
<td>Nancy Battle (NB)</td>
<td>10-15 years teaching experience. Attended Year 5 and Year 6 PD but did not teach Year 5 ScratchMaths. Worked closely with Year 5 SM teachers as preparation for Year 6.</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Christine Clark (CC)</td>
<td>Qualified teacher but works in a teaching assistant role, attended Year 6 PD. No previous Scratch or programming experience. Supports Nancy by teaching small groups of children in a separate classroom.</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>Barite</td>
<td>Matt Silver (MS)</td>
<td>&lt;5 years teaching experience. Taught Year 5 ScratchMaths, attended majority of Year 5 and Year 6 PD. Sole ScratchMaths teacher in school.</td>
<td>Place value and variable</td>
</tr>
</tbody>
</table>
Group A are characterised by having limited experience of teaching SM but had been teaching for at least three years. They had not taught SM in year 5 as not all primary teachers remain with their children at the end of the teaching year due staff changes and reallocation of teacher expertise across the school. Some of the group had attended SM professional development and engaged with the 5E pedagogical framework, and hands on experience with the microworlds. However, the scaffolded support over two days of PD could not replace the knowledge gained from teaching with the curriculum for a year. In some schools, such as at Humite and at Garnet, additional unplanned staffing changes meant that these teachers had missed the PD completely and yet were still expected to teach the ScratchMaths Curriculum e.g., Elizabeth, Betty and Trevor. These teachers did not engage with any knowledge transfer or training from the teachers who had taught the previous year. However, I was able to provide an additional one-hour discussion with Elizabeth and Trevor from Garnet a week before they were observed teaching ScratchMaths curriculum materials in the absence of the PD.

Group B had equivalent general teaching experience as Group A, but unlike Group A all the teachers had either taught SM to their class in the previous year (Matt, Laura and Katherine) or worked very closely with the Year 5 teachers in their school who had (Nancy and Christine).

3.4 Methods of data collection

The qualitative multiple-case study with augmenting teacher episodes approach required a range of data collection instruments and strategies to be developed. Some methods evolved in response to unplanned incidents or observations within the research, a direct consequence of the DBIR methodology. This section describes the data collections methods and links them to the research aims and questions.

3.4.1 Classroom observation

Classroom observation is more than just looking at what is taking place in the classroom. It is the conscious act of looking and noting systematically to gather information on teachers, children, events, behaviours, settings, routines and so on (Marshall and Rossman, 2014;
Simpson and Tuson, 2003). The distinctive feature of observation is that it offers the researcher the opportunity to gather, first hand, live data in situ in the real-life situation rather than for example using reported data (Wellington, 2015) or second-hand accounts (Creswell, 2018). The use of classroom observation as a principal method of research has the “potential to yield more valid or authentic data that would otherwise be the case with mediated or inferential methods.” (Cohen, Manion and Morrison, 2018, p. 542). As such, the use of classroom observations as the primary method of data collection provides the opportunity to gather rich contextual information that is strong on face validity; it can enable first hand data which may reveal verbal, non-verbal and physical aspects of teacher’s practices (Clark et al., 2009).

I used classroom observations as the primary data collection method to examine teachers’ Mathematical Pedagogical Technology Knowledge (MPTK) as they engaged through programming to teach and explore mathematical concepts. More specifically, teachers’ use of programming, teachers’ expressions of mathematics using the ScratchMaths curriculum materials and the teachers’ interplay between computer programming and mathematics. Given my focus of researching the teacher, rather than the learner, I adopted a position of observer-as-participant. In this role the children in the classroom were told that I was a SM researcher, and I would be as unobtrusive as possible, but also that I may participate a little in the groups’ activities. I adopted a non-structured approach to the observation itself, whereby the researcher is responsive to the agenda of the participant (the teacher) and the selectivity of what is observed is not dictated by the researcher’s observation schedule. Working in this way, allows key issues to emerge from the observation rather than the researcher knowing in advance what those key issues may be. Consequently, an unstructured observation provides a “rich description of a situation which, in turn, can lead to the subsequent generation and testing of hypotheses.” (Cohen, Manion and Morrison, 2018, p. 544). However, using an unstructured observation necessitates the need to capture and record information from the lesson observation as unambiguously and as faithfully and fully as is possible (Robson, 2002).

At the start of the data collection, I piloted recording the lesson using an audio recorder in addition to my observation notes. I selected an audio recorder to be as unobtrusive as possible, and to reduce the teacher’s concern of video capture. In my experience of observing beginning
teachers, the majority had no objection to an observer in the classroom but were increasingly concerned if they were to be video recorded. I used two audio recorders, one placed on the teacher’s desk to capture the teacher’s audio, and a second inserted into the teacher’s pocket to capture the audio from 1-1 teacher-pupil interactions. I also created systematic narrative and descriptive notes during the observation to aide later transcription and recall of the lesson. I included elements that would not be captured by the audio recording for example teacher’s use of digital technology and the white board, where the teacher was standing, the layout of the classroom as well as any significant teacher activity such as use of roleplay. My extensive experience of observing teachers’ practice as a mathematics tutor on initial teacher education programmes ensured I was attuned to notice teacher activity and able to create detailed written lesson observations.

However, reviewing the audio after the lesson revealed two technical challenges i) the audio recorders were not very successful at capturing the often inaudible very quiet pupil responses when the teacher worked 1-1 with children and ii) limited capturing of the teachers’ use of technology due to audio only. With respect to the second challenge, although the teacher would frequently announce and narrate what they were doing with technology, for example, “I drag out the next costume block from Looks”. When I reviewed the audio recording alongside my own detailed notes, I noticed gaps in what was happening on screen, since the teacher did not always narrate their actions. I evaluated that I had not captured a sufficient level of granularity of the teachers use of digital technology. I considered how to reduce or eliminate the technical challenges and responded quickly before the next scheduled observation.

The first challenge was managed by focusing on the observable (and consequently audio recording) of the whole-class teacher pupil interactions and dialogue, rather than teacher pupil 1-1 interactions. In the reviewed pilot observation, teachers’ 1-1 interactions with children were often rather brief and consisted of a repeat of whole-class instruction, rather than something specific to the individual child, or use of the computer. However, this limitation of only capturing whole-class interactions, would be addressed in the post-lesson interview if a 1-1 interaction was observed and noted in the unstructured observation record. For the second challenge it was imperative to be as invisible and impartial in the classroom and to capture the actions of the teacher’s use of Scratch on screen. Consequently, I decided to video record using
3.4 Methods of data collection

a tripod at the back of the classroom. Video recording can overcome the partialness of the observer’s view of the single event, and can offer a more unfiltered observation record in real time as it maintains the sequence of the events (Jewitt, 2012; Simpson and Tuson, 2003). Placing the video camera at the back of the classroom, behind the children, aimed to minimise the problem of reactivity, whereby children may react differently if video recorded. I decided to sit next to the camera when the teacher was teaching to the whole-class and would manually zoom to focus on the teacher’s interactive white board to capture detail when needed.

I also trialled using screen capture software of the teacher’s computer, but no teacher had any software preinstalled and due to school’s policies were unable to install any additional software their machine. In addition, some teachers did not allow me to video record for either personal preference or school policy, so I reverted to using the two audio recorders. To minimise the previously discussed issues, in these situations I aimed to capture richer detail in my observational notes, for example describing the teacher’s observable onscreen actions when they used Scratch, or when they involved children in roleplay or used children to operate the computer.

3.4.2 Teacher interviews

As well as classroom observation, I also used interviews to further examine teachers’ MPTK and to triangulate findings from the observations. My objective for the interviews was two-fold: i) to examine teacher knowledge of specific mathematical concepts, place value, and variable which may not have been directly observable within the lesson observation and ii) to examine and tease out teacher perceptions, intentions, and reflections of their practices which also may not have been directly observable within the lesson. The teacher interviews also provided me with an opportunity to follow up on any issues or question which arose during the lesson observation.

Kvale (1996) remarks that the use of interviews in research moves away from seeing data as external to the individual and more towards regarding knowledge as generated between humans often through conversations, in effect an inter-view or an interchange of views between people. He goes on to state that there are two different approaches to interviews, the miner who extracts
the information from the participant or the traveller who is concerned to travel with the participant into an unknown country. This latter position is one I have adopted, due to my research questions which focus on teacher’s MPTK being shaped and shaped by their engagement with teaching the ScratchMaths curriculum. Discussing knowledge at the intersection of mathematics will undoubtedly involve the co-construction of knowledge as the teacher’s discuss the lesson and the lesson content from their perspective, or how they regard the situation from their point of view. This will involve being an interviewer who is a traveller to a new country as Kvale describes.

I adopted a semi-structured interview protocol which allowed me the scope to discuss the observed lesson, but also to examine the specific concepts of place value, and variable in more depth, dependent upon the lesson content observed. In addition to the semi-structured interview questions, I also developed a place-value paper based task and a variable computer-based task, to examine teacher’s MPTK at a finer grained level which may not have been observable within the lesson observation. The design of these tasks is discussed in the next section.

### 3.4.2.1 Place value task

Hill, Schilling and Ball (2004) developed and empirically tested measures of teachers’ content knowledge for teaching primary mathematics using items that they developed and tested over five years. The survey items are underpinned by the knowledge that is needed to help students learn mathematics, specifically focused on the subject-matter knowledge that is required for teaching. The original subset of items was presented in a multiple choice format and readers of the paper were strictly advised that the items should not be used as an overall measure by calculating the number of correct responses per teacher or to gauge growth over time. However, the authors stated that the items could be used as open-ended prompts for discussion about exploration of teachers’ reasoning about mathematics and students thinking. For the semi-structured interview, I therefore selected three items from the published set of 35 which focussed on teachers’ mathematical content knowledge of place value. I adapted the questions to be open by removing the multiple choices and included additional interviewer prompts and probes. The questions are shown in Table 3.2 below.
The second part of the semi-structured interview was designed to examine teachers’ MPTK at the intersection of mathematics and computing, specifically focussed on the teacher’s use of the ScratchMaths place value microworld. Questions 4, 5, 6, 7 as shown in Table 3.2 asked the teacher to recall the place value model they had been teaching with. The SM Place value microworld is used in all investigations from SM Module 4. (The SM Module 4 - Building with numbers and Place value microworld is described in detail in Chapter 4). The questions were adapted from a task which I co-designed and discussed in more detail in a separate publication (Benton et al., 2018b). I added Questions 8, 9, 10 to illicit further detail on the teacher’s use of ScratchMaths to reveal any behaviours or practices which may not have been observable in the lesson observation, by asking the teacher to reflect on their pedagogical practices.
Table 3.2 Place value mathematical content knowledge items used in teacher interviews

<table>
<thead>
<tr>
<th>Qu.</th>
<th>Mathematical content item</th>
</tr>
</thead>
</table>
| 1   | Children are working on putting decimals in order.  
Three students wrote 1.1, 12, 48, 102, 31.3, .676 ordered from least to greatest.  
*What error are these students making?*  
**Prompt:** Is this a type of error that you have seen?  
**Probe:** What would your intervention be? |
| 2   | A teacher is reviewing children’s work on column addition. She sees the following three student mistakes:  

<table>
<thead>
<tr>
<th></th>
<th>I)</th>
<th>II)</th>
<th>III)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38</td>
<td>45</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>37</td>
<td>14</td>
</tr>
<tr>
<td>+ 65</td>
<td>+ 29</td>
<td>+ 19</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>101</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

*Which answers have the same kind of error? Explain your reasoning.*  
**Probe:** If this is the only pupil’s work, which pupil has the ‘best’ understanding of the algorithm? |
| 3   | Working with a child, you ask her to count out 23 counters, which she does successfully. You then ask her to show you how many counters are represented by the 3 in 23, and she counts out 3 counters. You then ask her to show you how many counters are represented by the 2 in 23, she counts out 2 counters.  
*What problem is the child having?*  
**Prompt:** Is this a type of error that you have seen?  
**Probe:** What would your intervention be? |
### Methods of data collection

<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.</strong> How would you teach add 4 to 399? In scratch?</td>
<td>I want to adapt the place value model to represent time. How should I do this? Explain how the time model connects to the place value model. <strong>Prompt:</strong> What would the sprites be called? <strong>Probe:</strong> During Module 4, in what ways do you exploit the children’s prior knowledge in maths? <strong>Prompt:</strong> What is the point of constructing the place value model? <strong>Probe:</strong> How do you bridge from computing to mathematics? Explain how you will make connections between the place value ScratchMaths model and place value when teaching.</td>
</tr>
<tr>
<td><strong>5.</strong> I want to extend the model to show numbers above 99. I suggest the first thing I would need to do is add another sprite (by duplicating). How should I do this? <strong>Prompt:</strong> Where would the sprite go, what should I call it? <strong>Probe:</strong> What could I do to make sure the model works?</td>
<td><strong>6.</strong> I want to extend the model to show numbers above 99. I suggest the first thing I would need to do is add another sprite (by duplicating). <strong>Prompt:</strong> Where would the sprite go, what should I call it? <strong>Probe:</strong> What could I do to make sure the model works?</td>
</tr>
<tr>
<td><strong>7.</strong></td>
<td><strong>8.</strong></td>
</tr>
</tbody>
</table>
3.4.2.2 Variable computer-based task

For the post-lesson interviews of observed lessons for ScratchMaths Module 5, I developed structured reflection questions and a computer-based task (Question 8) as shown in Table 3.3. Goldin (2000) argued that task-based interviews can serve as a research instrument to study the mathematical behaviour of a problem solver but can also allow the researcher to describe the subject’s knowledge of mathematics. However, my use of a computer-based task is to provide another opportunity to open a window into teachers’ MPTK in addition to that of the lesson observation. The task-based interview would enable me to trace teachers’ mathematical knowledge as they learn to program in Scratch using a task which focused on using variable in mathematics and computing contexts. Specifically, questions 1 and 2 focused on teachers’ content knowledge of the term variable. They were asked to define the term and to consider what they would say if they were asked by a child in their classroom what a variable is. Questions 3-5 examined teachers’ content knowledge of variable in the specific contexts of computing through their use of Scratch in the SM modules. Question 6 changed the context to mathematics and examined teacher’s MPTK to bridge between the computing and mathematics. Question 7 examined the teacher’s MPTK of variable as expressed in Scratch through the creation of regular polygons of fixed sizes.

The task-based question 8 provided a task to explore teachers’ MPTK of variable in mathematics and variable in computing through use of a Scratch based task which incorporated geometrical ideas. The task is an adaptation of a pupil task from Module 5 (see Chapter 4) where the teachers are asked to use Scratch to generate the output shown in the images and explain their thinking at each step. The task was designed so that it would provide an opportunity to examine how teachers’ conceptualised variable in computing and mathematics, since a solution to the task requires the subject to engage with variable represented in computing as generalised number in mathematics (see section 2.3.2).

I observed Rina six times, Sally four times and Mary eight times. All other teachers from the teacher episodes were seen once for either module 4 or 5. Each observation was followed by a post-lesson interview, either immediately, or scheduled for the next day, using either the place value task, the variable task, or a more general interview based on discussing items I had
captured in my observation notes. The full detail of the visit schedule is shown in Table 10.2 of Appendix H.
<table>
<thead>
<tr>
<th>Qu.</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1   | Thinking about the word ‘variable’, where did you first encounter it?  
*Probe:* In study/school/uni, self-study, online, in which discipline/subject? |
| 2   | What is your understanding of variable?  
*Prompt:* What is your definition? Can you give me an example, if a definition is hard. What would you say to a pupil if they asked “Miss/Sir, what is a variable?” |
| 3   | Working in Scratch, can you think where we have seen or used variables?  
*Probe:* Can you recall any moments when you thought, we are using a variable here! |
| 4   | How does ask and answer fit into the concept of variable?  
*Probe:* Are they the same thing, or are they different things? |
| 5   | If you think about costume #, is that a variable?  
*Probe:* How do you know? |
| 6   | How does your understanding of variable in computing bridge to mathematics?  
*Probe:* Can you think of an example in mathematics which uses variable? Is this something you have used with Year 6? |
| 7   | Imagine the situation: a child has successfully created a script for a square, a pentagon and a hexagon.  
*What would you ask the child to do next? Explain the rationale for your choice.*  
*Probe:* What might the child find difficult? What support would they need? What would you look for in their work? |
### 3.4 Methods of data collection

A child has built a polygon script which asks for the number of sides, and the side length. The output is shown below.

Using Scratch (if available) build what the script might look like. Explain your thinking.

**Probe:** What are the important features of the script to look for?

How would you assess their work from a mathematical perspective?

What does their work demonstrate they understand?

---

<table>
<thead>
<tr>
<th>8</th>
<th>A child has built a polygon script which asks for the number of sides, and the side length. The output is shown below.</th>
</tr>
</thead>
</table>

*Table 3.3 Variable task and reflection questions used in teacher interviews*
3.4.3 Labelling and managing data

A systematic approach to data labelling and data reporting is used throughout this study. The teacher and school can be identified from the three characters assigned to them. The first character identifies the school and the second and third character identify the teachers first and last name. All names are pseudonyms. For example, data related to Mary Waters of Diamond Primary School would be identified by the three characters DMW. See Table 3.4 for the complete set of codes.

<table>
<thead>
<tr>
<th>Local area</th>
<th>School (Code)</th>
<th>Teacher</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradford</td>
<td>Amber Primary School (A)</td>
<td>Laura Havers</td>
<td>ALH</td>
</tr>
<tr>
<td>Merseyside</td>
<td>Barite Primary School (B)</td>
<td>Matt Silver</td>
<td>BMS</td>
</tr>
<tr>
<td></td>
<td>Carnelian Primary School</td>
<td>Katherine Williams</td>
<td>CKW</td>
</tr>
<tr>
<td>North London</td>
<td>Diamond Primary School (D)</td>
<td>Mary Waters</td>
<td>DMW</td>
</tr>
<tr>
<td></td>
<td>Emerald Primary School</td>
<td>Rina Etherington</td>
<td>ERE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sally Edwards</td>
<td>ESE</td>
</tr>
<tr>
<td></td>
<td>Freshwater School (F)</td>
<td>Priya Rashmi</td>
<td>FPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anthony Oliver</td>
<td>FAO</td>
</tr>
<tr>
<td>South London</td>
<td>Garnet Primary School (G)</td>
<td>Elizabeth Angel</td>
<td>GEA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trevor Trafford</td>
<td>GTT</td>
</tr>
<tr>
<td></td>
<td>Humite Primary School (H)</td>
<td>Betty King</td>
<td>HBK</td>
</tr>
<tr>
<td>Blackburn</td>
<td>Jolite Primary School (I)</td>
<td>Nancy Battle</td>
<td>INB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Christine Clark</td>
<td>ICC</td>
</tr>
</tbody>
</table>

Table 3.4 Table of codes by local area, school and teacher

In addition, the range of different types of data that contributed to the research and the case studies required a second level of identification to be developed, a modified version of codes used by Clark-Wilson (2010). The data labelling scheme adopted throughout this thesis is shown in Table 3.5.
3.4 Methods of data collection

<table>
<thead>
<tr>
<th>Data source</th>
<th>Code</th>
<th>Example in thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcript of lesson observation</td>
<td>Obvs</td>
<td>DMW1(Obvs)</td>
</tr>
<tr>
<td>Interview with teacher</td>
<td>Int</td>
<td>DMW1(Int)</td>
</tr>
<tr>
<td>Digital image of the teacher’s white board</td>
<td>Image</td>
<td>DMW1(Image1)</td>
</tr>
<tr>
<td>Video of lesson</td>
<td>Video</td>
<td>DMW1(Video)</td>
</tr>
<tr>
<td>Scratch project developed by the teacher for use by the teacher</td>
<td>Scratchn</td>
<td>DMW(Scratch1)</td>
</tr>
<tr>
<td>Teacher’s response to Place Value Task by question</td>
<td>PVT-Qn</td>
<td>DMW(PVT-Q1)</td>
</tr>
<tr>
<td>Personal journey containing meeting notes, notes following observations, email communications, small snippets of conversations following lessons not part of a formal interview.</td>
<td>Journ</td>
<td>Journ</td>
</tr>
</tbody>
</table>

Table 3.5 Data source codes used in thesis

3.4.4 Challenges during data collection

During the final year of the ScratchMaths intervention (2016-2017), collecting data from the Year 6 teachers across the country who were part of the intervention, with a reliable consistency, was extremely challenging. Many participant teachers cancelled their ScratchMaths lessons as their schools turned their attention to preparing children for the KS2 examinations. Anything that was deemed additional to core practice in Maths and English was cancelled. This was disappointing given the hypothesis that if ScratchMaths was implemented as designed, it has the potential to improve pupils’ mathematics. However, this can partially be explained since an untested and new approach to a curriculum vs traditional examination practice is a difficult position for head teachers to take in the face of parental pressures and external accountability. An additional barrier to collecting data was the physical distance from myself and the participants. I was unable to organise an observation with several of the teachers despite their indication to be part of the study when they were met face to face at the Year 6 Professional Development. Some teachers were keen, but would not formalise when I could visit, despite repeated requests. The teachers’ reason was unfamiliarity with the materials and wanting time to develop confidence before being observed. However, when I followed up later in the school year I was met with silence or excuses due to examination practice. This was an
unfortunate outcome since all teachers involved in the ScratchMaths intervention had signed a memorandum of understanding (MOU) during their PD which included a clause (see Appendix G) to be part of research projects and to allow researcher observations.

However, as previously discussed, the three teachers, Mary, Rina and Sally re-scheduled their lessons and I continued to work with them over the year. Their data formed the three cases presented in Chapter 5, 6, 7. The additional data collected from the teachers at other schools, who were seen lesson frequently and at different points of the school year, is presented as in Chapter 8 as a series of teacher episodes.
3.5 Operationalising my framework for teaching mathematics through programming in the primary setting

This study aims to identify what mathematical, and computing teaching knowledge is required to teach ScratchMaths activities on place value and variable effectively. The conceptual framework that I have developed for analysing teachers’ MPTK for teaching mathematics through computing in the primary setting was discussed in section 2.4.3 of Chapter 2. The conceptual framework extends Thomas and Palmer’s (2014) original PTK framework and includes Clark-Wilson and Hoyles’ (2017) amendment of a bi-directional arrow. A teacher’s MPTK incorporates the principles, conventions, and techniques required to teach mathematics through programming. The theoretical framework has the following components:

- **Pedagogical knowledge:** A teacher’s knowledge of the principles and strategies of classroom management.
- **Mathematical content knowledge:** A teacher’s own knowledge of mathematics.
- **Computational content knowledge:** A teacher’s own knowledge of computing.
- **Mathematical/Computational Knowledge for Teaching:** The mathematical and computational knowledge alongside the pedagogical knowledge needed to perform teaching of tasks from the ScratchMaths curriculum.
- **Personal orientations:** The teachers’ goals, attitudes, dispositions, beliefs and their perceptions of the nature of mathematical knowledge and how it should be learned with or without technology.
- **Technology instrumental genesis:** The process by which the teacher’s mathematical and computational knowledge shapes and is shaped by their interactions with Scratch as they accomplish a particular ScratchMaths task. This also incorporates the teacher’s understanding of the development of the children’s processes of instrumental genesis, as they engage with Scratch to explore mathematical ideas.
The MPTK framework was operationalised as shown in Table 3.6. The table how each component of the framework was researched within the study and which of the data collection methods were used.

<table>
<thead>
<tr>
<th>Component</th>
<th>Construct</th>
<th>Method</th>
</tr>
</thead>
</table>
| Mathematical/Computational Knowledge for Teaching | Pedagogical Knowledge | • Lesson observation  
• Place value task-based interview  
• Variable task-based interview |
| Mathematical Content Knowledge | | • Lesson observation  
• Place value task-based interview  
• Variable task-based interview |
| Computational Content Knowledge | | • Lesson observation  
• Place value task-based interview  
• Variable task-based interview |
| Personal Orientations | | • Lesson observations and interviews |
| Technology instrumental genesis | | • Lesson observation  
• Place value task-based interview  
• Variable task-based interview |

Table 3.6 Mapping data collection methods against the MPTK framework

On their own the different components of Thomas and Palmer's (2014) original PTK framework provides categories of teacher knowledge but does not exemplify how the model can be utilised to examine teacher’s knowledge. However, Clark-Wilson and Hoyles (2017) in their exploration of teachers’ MPKT in the Cornerstone project identified several processes involving the use of representations which they felt a teacher might enact or describe in a lesson when doing the “mathematical work of teaching” that involved the use of mathematical and technological representations. Utilizing this approach for my own research to perform the fine-grained analysis that I required to answer my research questions, I elaborated my MPTK framework a priori. I characterised each of the different components of MPTK which might be observable in a lesson and result in a teacher’s strong adherence to the original design of a ScratchMaths lesson underpinned by the 5E pedagogical framework as show in Table 3.7 below.
<table>
<thead>
<tr>
<th>Aspect of the MPTK framework</th>
<th>Lesson features</th>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedagogical Knowledge</td>
<td>Uses questioning strategies</td>
<td>The teacher develops and uses open and closed questioning strategies to encourage children to explain their thinking. The teacher organises classroom activities to use multiple activity types such as individual work, paired discussion, small group work or whole-class discussions. For example, the teacher poses a question for children to discuss in pairs for 30 seconds, and then selects pairs to feedback their discussion to the rest of the class.</td>
</tr>
<tr>
<td>Mathematical Content Knowledge</td>
<td>Refers to mathematical concept (i.e., place value, variable, angle.)</td>
<td>The teacher uses mathematical vocabulary to discuss place value, for example base 10, place value names and how places are related to each other e.g., adjacent places are 10 times bigger or smaller. The teacher uses place value charts, or other tables and diagrams to represent place value. The teacher may use manipulatives such as Diene’s apparatus or counters to represent place value problems. The teacher uses mathematical vocabulary to discuss algebra and variable, for example the teacher uses and defines mathematical terms such as variable, expression or discusses how an unknown quantity can be represented using a word or a letter. The teacher uses diagrams and labels them with words or symbols to represent unknown quantities.</td>
</tr>
<tr>
<td>Computational Content Knowledge</td>
<td>Refers to computational concept (i.e., loop, variable, algorithm, broadcasting)</td>
<td>The teacher uses and explains computational concepts such as decomposition i.e., solving a problem by decomposing it into smaller parts. The teacher uses the terms sequence, selection, repetition, logical reasoning, algorithm and provides examples. For example, the teacher represents an algorithm as a series of steps which are executed in sequence and defines the term variable in the context of computing.</td>
</tr>
<tr>
<td>Technology instrumental genesis</td>
<td>Uses whole-class technical orchestrations</td>
<td>The teacher uses Scratch in a live setting and describes their actions of the mouse and the keyboard, an example of Technical-demo. The teachers drag out a move block from the motion Scratch palette and describes that the sprite will move a certain number of steps. They explain where in the scratch interface the move block can be found. The teacher then edits the hole in the block to be 30 and narrates that they have typed 30 on the keyboard. The teacher encourages children to explain and to explore aspects of Scratch. For example, children are asked to snap a move and a turn block together and to explore the output on the screen when the script is run multiple times. The teacher uses and encourages children to use Scratch specific vocabulary such as costume, broadcast, block, ask and answer, stage, panel etc.</td>
</tr>
</tbody>
</table>
This section brings together the knowledge components of Mathematical and Computational Content Knowledge with Pedagogical Knowledge. To exemplify I continue the previous example.

Following the teacher’s actions of adding the move 30 block to the script, the teacher summarises that the block will make the sprite move 30 steps in the direction that it is currently facing. Finally, the teacher runs the script and explicitly connects the movement of the sprite on the screen with the corresponding block in the computer program, an example of bridgE.

<table>
<thead>
<tr>
<th>Mathematical/Computational Knowledge for Teaching</th>
<th>Personal orientations</th>
<th>Mathematical Pedagogical Technical Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses questioning strategies</td>
<td>Positive approach to using technology.</td>
<td>Uses whole-class orchestrations</td>
</tr>
<tr>
<td>Refers to mathematical and computational concepts</td>
<td>Mistakes are discussed and encouraged.</td>
<td>Uses questioning strategies</td>
</tr>
<tr>
<td>Describes acting on/connecting mathematical or computational representations</td>
<td></td>
<td>Refers to mathematical and computational concepts</td>
</tr>
<tr>
<td>Uses mathematical and computational vocabulary</td>
<td></td>
<td>Uses mathematical, computational and Scratch vocabulary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Describes acting on/connecting mathematical or computational representations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supports children in their developing instrumental knowledge of Scratch</td>
</tr>
</tbody>
</table>

The teacher views technology as supportive to children’s learning experiences.

The teacher uses their own mistakes and children’s mistakes as opportunities for learning.

The teacher encourages children to exchange their ideas. For example, the teacher selects a child’s Scratch project to display on the screen and asks the class to identify bugs within the project. The children discuss the project in pairs, which might include copying parts of the script to explore on their own computers, and then to explain why the bug is present and how they might fix it. The teacher holds a whole-class discussion to debug the project, an example of Discuss-the-screen.

The teacher explains how the place value structure represented in Scratch corresponds to the place value structure in mathematics, an example of Explain-the-screen or an example of bridgE. The teacher asks the children to envisage what would happen when the ones sprite is clicked one more time in the Scratch place value model if the sprite is currently wearing the costume of 9.

Table 3.7 A model for operationalising the MPTK framework to examine teachers’ classroom practice using the ScratchMaths curriculum
3.6 Data analysis

Qualitative data analysis is a range of processes by which a researcher moves from the qualitative data collected, into some form of understanding, explaining and interpreting of the phenomena in question (Lewins, Taylor and Gibbs, 2010). As with research design, qualitative data analysis is not straightforward. There is no one single or correct way to approach it, but the process should also abide by having a fitness for purpose (Cohen, Manion and Morrison, 2018). In the next section I outline the analytic approach I developed for the purpose of addressing my research questions.

3.6.1 Developing an analytical approach for data analysis

I started preliminary data collection in May 2016 during the Year 6 ScratchMaths professional development event which I delivered nationally to approximately 50 teachers in different parts of the country. These events were positioned optimally before the start of the school year (the second year of the ScratchMaths intervention). Data collection of the lesson observations and task-based interviews began at the start of the school year in September 2016 and continued until July 2017 (the end of the ScratchMaths intervention). However, case teachers Mary, Rina, and Sally, continued to teach ScratchMaths into the next school year thus providing additional data collection opportunities. I finished all data collection in March 2018.

The analytic approach I followed:

i. Creation of a lesson observation script for each observation. I listened to the audio and watched the video recording of each lesson. All whole-class audio activity of the teacher and pupil interacting was transcribed word for word. Timestamps were coded at the start of a single teaching episode defined as when the teacher starts a new task within the lesson or moved between teaching activity, or children started independent work. I inserted photos taken from the board during the lesson or screen grabbed directly from the video feed when the teacher was using technology. If the teachers’ focus was Scratch code displayed on the screen and
had been poorly captured by the video recorded, I recreated the teacher’s screen in Scratch and took a screen shot. I added detail to the script, noting the teachers’ actions with technology. For example, “She created a variable block, named it side length and dragged it onto the stage.” I included commentary and additional notes of observations made during the lesson, e.g., when children had their hands up or teachers’ actions during pupil independent work. An example is shown in Appendix B.

ii. Creation of a lesson observation spreadsheet. The lesson observation script was transformed into a lesson observation spreadsheet. The purpose of the spreadsheet was to create a visual representation of the lesson at a level of abstraction higher than the detailed lesson script. Each episode, defined as a fragment of the lesson concerning one task and one type of whole-class orchestration (Drijvers et al., 2013) was inserted as a row into the spreadsheet. The episode is time stamped at the start of each episode which allows the length of the episode to be calculated. A sample quote was inserted for easy locating within the lesson observation script. An example is shown in Appendix A.

iii. Coding of the lesson observation spreadsheet through the lens of the Technology instrumental genesis component of the theoretical model. Coding of individual teaching episodes within the observation spreadsheets was performed using codes named as the whole-class orchestrations types developed by Drijvers et al. (2010); Drijvers et al. (2013)

iv. Creation of an Interview script. Each interview audio recording was transcribed word for word into a document. For the interview-based tasks, screen grabs of code were inserted into the script, alongside descriptions of the teacher’s activity in the microworld.

v. Coding of the observational data through the lens of the Mathematical / Computational Knowledge for Teaching component of the theoretical model. Following the coding of the lesson observations using the whole-class orchestrations with a primary focus on technology use, the observations were analysed through the lens of Mathematical Knowledge for Teaching to examine the interplay between mathematics and expression of mathematics using
technology. To obtain this level of granularity, each observation script was reread, looking specifically for expressions of mathematics, or mathematical pedagogy whether in the teacher’s language or in their use of Scratch or other representative forms. Using a constant comparative strategy (Corbin and Strauss, 2014) across all of the teachers, I identified emerging themes and coded the lesson observation scripts based on the elements of the theoretical model using NVivo 11, a computer-assisted qualitative data analysis software which supports analysis through textual searching and creation of summary tables of nodes.

vi. Coding of the interview data and interview-based tasks through the lenses of Mathematical Knowledge for Teaching and Technology instrumental genesis components of the theoretical model. I reviewed the transcripts of the interviews in two distinct passes, first using a lens of mathematical knowledge for teaching and second using a lens of technology instrumental genesis. Again, using a constant comparative strategy, I identified additional themes in both passes and coded using NVivo 11.

vii. Validity - triangulation of the data: I triangulated the data collected from the lesson observations with the data collected from the interviews/task-based interviews to address internal validity. For example, to triangulate teacher’s mathematical knowledge for teaching place value, the collected data from the lesson observations was compared with the teacher’s responses to the open prompt place value discussion items. Similarly, to examine teacher’s Mathematical / Computational Knowledge for Teaching from the theoretical model, the data collected from the lesson observation was compared with the data collected as the teacher’s worked on the Scratch based variable task in the interview.

viii. Inter-coder reliability – assessing researcher coding bias. After completing the analysis of the lesson observations and interviews for one of the case study teachers. I asked another member of the ScratchMaths research team, who had experience in collecting and analysing qualitative data, to follow data analysis step iii) to code a lesson using the whole-class orchestration (Drijvers et al., 2013). We
then met to compare our analyses and found that our analysis had produced near identical coding of teaching episodes. Where there was a difference in codes of application of codes, for example, in one episode they coded “Explain-the-screen” (which has a mathematical focus), rather than “Discuss-the-screen” where the mathematics is less prevalent. We agreed that the episode was a borderline example and could be coded either way and would therefore not impact overall characterisation of the teachers’ practices.

ix. **Validity – member checking.** The objective of member checking or participant validation is to find out if “the data analysis is congruent with the participants’ experiences” (Curtin and Fossey, 2007, p. 92). Validation from the participants would therefore enhance the trustworthiness and hence validity of my analysis and findings. I emailed each case study teacher their polished case analysis chapter. I asked each teacher to read their chapter and to capture any comments where they felt that I had been expressed them inaccurately, or if I had written anything that they objected to. All three main case study teachers completed this process which did not lead to any changes in the analysis or of the case study presentation.

### 3.6.2 Piloting the analytical approach

The analytical approach described above suggested I followed a linear sequence of coding. However, in conducting the analysis, my approach was much more iterative. For example, in step iii) **Coding of the lesson observation spreadsheet through the lens of the Technology instrumental genesis component of the theoretical model,** I coded the whole-class orchestrations of two observed lessons on the lesson observation spreadsheet. Out of the 8 teaching episodes identified and coded within the two lessons, the most frequent orchestration coded (see Table 3.8) was **Board-instruction,** the catch-all orchestration used by Drijvers et al. for the traditional activity of a teacher whole-class teaching in front of the board without the use of technology.

<table>
<thead>
<tr>
<th>Code</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board-instruction</td>
<td>4</td>
</tr>
<tr>
<td>Discuss-the-screen</td>
<td>1</td>
</tr>
<tr>
<td>Explain-the-screen</td>
<td>3</td>
</tr>
</tbody>
</table>
3.6 Data analysis

<table>
<thead>
<tr>
<th>Guide-and-explain</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link-screen-board</td>
<td>0</td>
</tr>
<tr>
<td>Sherpa-at-work</td>
<td>0</td>
</tr>
<tr>
<td>Spot-and-show</td>
<td>0</td>
</tr>
<tr>
<td>Technical-demo</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.8 Summary of code frequency after pilot coding two lesson observations

However, these were lengthy sequences within each lesson and represented a large proportion of activity that involved two different teacher orchestrated activities i) the use of digit flip books to model the mechanics of sprites’ costumes in a place value model and ii) an activity which involved moving cards into different sections of a board for a place value game. Both activities were unplugged, meaning activities away from the computer but precursors to children implementing and programming in Scratch. The teacher made links between different modes of representation of mathematical and computational ideas e.g., the flipping of a digit card to represent adding one in mathematics, and changing costume in scratch, the mechanic of nudging in the physical flip game and checking a conditional statement in the mathematical context of place value and the corresponding Scratch language of broadcasting and receiving. These activities are not typical of traditional teaching as in Board-instruction nor could be described by any of the other categories such as Discuss-the-screen or Explain-the screen, I therefore created a tentative additional whole-class orchestration category labelled Link-multimodal-representation to capture such activity and used the additional category to recode the two observed lessons, before continuing with the rest of the coding.

3.6.3 Issues of validity

During data analysis I returned repeatedly to the question of the validity of my methodological approach by viewing the observation data through multiple lenses when coding against my theoretical model. In Step iii) of section 3.6.1 above - Coding of the lesson observation spreadsheet through the lens of the Technology instrumental genesis component of the theoretical model, I used the whole-class orchestration codes as one component to examine teacher’s Mathematical Pedagogical Technology Knowledge. For example, exemplifying the whole-class orchestration Discuss-the-screen whose goal is to enhance collective instrumental genesis:
- If a teacher displayed a ScratchMaths curriculum material project on the screen, edited the script live and asked open questions of the script, I coded this as *Discuss-the-screen*,
- If a teacher displayed a ScratchMaths project on the screen, but used their own broken project, one they had specially designed to debug live as a class, this was also coded as *Discuss-the-screen*,
- If a teacher used Scratch live, asked discussion questions, edited the script according to responses from the children, but were not always able to explain the final output on the screen, this was also coded as *Discuss-the-screen*.

There was clearly a continuum of teacher’s use of the *Discuss-the-screen* that was failing to be captured by the single code. The teacher’s use of the different types of whole-class orchestrations is an important aspect of their pedagogical approaches but does not sufficiently capture how they are using the specific orchestration.

The same issue arose in step v) of section 3.6.1 above *Coding of the observational data through the lens of the Mathematical / Computational Knowledge for Teaching component of the theoretical model*. When a teacher used one of the 5 E’s from the 5E pedagogic framework, I coded the episode using the appropriate E. However, the codes themselves did not provide enough nuance for describing the depth of the teacher’s pedagogical approaches, or their engagement with the ScratchMaths curriculum materials. For example, a teacher could encourage the children to *Explore* an aspect of a ScratchMaths project, without necessarily being able to respond to the children’s explorations. Similarly, a teacher could encourage the children to *Explain* an aspect of a script displayed on the IWB, or to *Explain* the mathematical content, but the teacher’s use of *Bridging*, between context was present, but limited by how they responded to children’s explanations and possible misconceptions. I realised that by coding in this way and focusing on the specific components of the theoretical model holistically across all observed lessons I was losing an aspect of the teacher’s evolution of knowledge over time. My use of themes could describe how a teacher used a specific strategy, or engaged with mathematics, but by themselves they did not holistically describe how these pieces come together in an individual lesson and provide opportunities for a child to engage deeply with mathematics. In order to provide such a holistic view, I applied an analytic tool for lesson
analysis called *The teaching for robust understanding (TRU) framework* (Schoenfeld, 2013; Schoenfeld, 2014). Schoenfeld’s framework was selected because i) it would capture what takes place in mathematics classrooms, ii) it focused in clear ways on dimensions of classroom activities that were known in the literature to be important and not captured in Thomas and Palmer’s PTK framework, and iv) from prior studies the scheme demonstrated reliability and validity. The tool is discussed in the next section.

### 3.6.4 Teaching for robust understanding in the context of teaching mathematics through programming in the primary setting

The complexity of constructing a classroom analysis scheme for empirical use was addressed by Schoenfeld in the development of the teaching for robust understanding (TRU) framework (Schoenfeld, 2013; Schoenfeld, 2014b). Schoenfeld defined robust understanding in the context of students’ abilities to be effective at dealing with, making sense of, and solving contextually rich tasks (Schoenfeld, 2013). For this study, the context of the framework is using mathematics concepts in varied, new and challenging situations, such as through computer programming. The TRU framework has the core assertion that the activities that take place in a classroom can be considered as a taking place in a metaphoric five-dimensional space. The five dimensions of classroom practice as shown in Figure 3.1 below are the necessary and sufficient to characterise the kinds of teaching for effective classroom instruction, if the goal is for “students are to emerge from that classroom being knowledgeable and resourceful disciplinary thinkers and problem solvers.” (Schoenfeld, 2018, p. 491). The framework is built upon more than a decade of research of models of teaching, and is grounded in the literature concerning mathematics teaching and learning as represented in the mathematics handbooks of research (e.g. see English (2008) and Lester (2007)). Importantly, and of interest to this study, is that the framework can be applied systematically to video analysis and snapshots of teacher’s practice (Schoenfeld, 2018). Applying the framework in a rigorous way would address the issues of reliability validity I raised in the previous section and would provide a snapshot of the lesson and the teacher at a particular point in time.
Schoenfeld initially presented The TRU framework as applying to all forms of mathematical instruction, however the framework has also been generalised by other researchers for use in other subjects such as science and English language, however these remain unpublished. An example TRU maths rubric which a researcher can use to measure mathematics classroom performance along the five dimensions of classroom activity is published at the Mathematics Assessment Project website (Mathematics Assessment Project, 2021). To ensure that I had interpreted the framework methodology correctly and could proceed with adaptations for the specific context of teaching mathematics through programming, I engaged in an email conversation with Schoenfeld. Schoenfeld confirmed that the expectation of the framework was that “there would be a lot of additional detail to flesh out in each dimension, with context-specific adaptations where appropriate.” (Schoenfeld, 2020). As a result of the email conversation, I modified the mathematics scoring rubric (Schoenfeld, 2014a) that I would use
to analyse the ScratchMaths lessons. I adapted the criteria for Dimension 1: The content at each of the three levels of the rubric i.e., Basic, Proficient and Distinguished. Each rubric descriptor statement was then adapted to include the teachers’ explicit use of the 5E pedagogical framework of the ScratchMaths curriculum and their explicit use of whole-class instrumental orchestrations (see Appendix A for all descriptors of the scoring rubric). Both activities are aspects of effective classrooms for teaching with technology and teaching using the ScratchMaths curriculum. For example, the Basic criteria for dimension 1:

Classroom activities are unfocused or skills-oriented, lacking opportunities for engagement with key grade level content (as specified in the Common Core Standards)

Became:

The teacher makes limited or no use of technology, simply tells students what to do with no clear aim in mind. Student computer activity is unstructured, unfocused and with little or no use of the 5E’s.

The adapted TRU scoring descriptors used many of the key words from the initial descriptors of Schoenfeld’s TRU framework but was changed the to the context of the ScratchMaths curriculum by using the 5E pedagogic framework. However, unlike the 5E pedagogic framework which would be known by the teachers if they used the ScatchMaths curriculum materials and attended the professional development, the adapted TRU framework was not known as it was developed during data analysis as an analytical tool. Thus, the use of the framework for developmental conversations with teachers was beyond the scope of this research but could be of interest for further study.

The process I used to analyse by scoring each lesson observation across the five dimensions is outlined below and described in Schoenfeld (2018):

i) I broke up each lesson into episodes of approximately 10 minutes’ length. If there was a natural break, for example the class moved from whole-class to paired work that would define the end and start of an episode. If the same activity went on for more than 15 minutes, I found a natural break point and created a new episode at the break point.
ii) I gave each episode a score of 1 (basic) to 3 (distinguished) using 0.5 intervals of a 5-point scale for each dimension using the language from the adapted rubric shown in Appendix A. Meanings were interpolated for scores of 1.5 and 2.5 from the descriptors of the rubric of scores 1, 2 and 3.

iii) I gave credit for activities that supported higher scores. For example, within an episode, the teacher did not have to be a consistent 3 to get a score of 3 but they had to have showed enough 3ish behaviour for me to be able to say it was present.

iv) Since episodes were of different length, I created a summary score for each dimension by using a weighted average, weighted by time. (See Appendix E for an example calculation).

v) Inter-coder reliability – I asked the same researcher who coded my lesson observations to code the lesson observations using the adapted TRU rubric. Our scoring in each dimension was at most 0.5 apart, in-line with findings from Schoenfeld (2018) when working with a research team of ten members, which suggested good inter-rater reliability of using the rubric.

3.7 Ethical considerations

Ethical issues in qualitative research need to be examined as they apply to the different phases of the research process i.e. before the study, during data collection, data analysis and reporting the data (Creswell, 2018). Although I was part of the ScratchMaths project team who had ethical approval, and consent from the participants to engage in research, it was necessary for me to create a separate doctoral student ethics application form to the University Collect (UCL) Research ethics committee. My ethical approval was granted by the UCL Institute of education (see Appendix F for the ethics form). The form adheres to the ethical guidelines as specified by the British Educational Research Association (2011).

All headteachers at participating schools received and signed a memorandum of understanding (MOU) as part of the ScratchMaths project. The MOU (see Appendix G) details the nature of the project and intervention and states that head teachers should allow the ScratchMaths project team to visit and observe lessons and to allow participating staff to take part in, for example,
3.7 Ethical considerations

research interviews, surveys and events within reasonable scope of their time and availability. Given that I was a ScratchMaths project team and researcher for my PhD, I additionally provided an overview of my planned research at the start of the Year 6 professional development sessions and informed the teachers that I would be in contact at the start of the calendar year to plan the first observation. In a follow up communication following the PD sessions, I explained to the teachers the processes of data collection for the observations the interviews and reminders about how I would protect teacher and school confidentiality in any reporting of the data such as in this thesis, using pseudonyms. As a final step, I reconfirmed teachers’ verbal consent at the start of the lesson observation or interview to record audio or video as appropriate. However, in some cases, as I discussed earlier, video consent was not given, due to either teacher confidence or school restrictions, and the teacher’s provided their consent for audio recording instead. All video recordings of the classroom were conducted in line with the school’s policy on video recordings. For example, although my video recording was focused on the teacher and the IWB at the front of the classroom only, there was potential for the back of a child’s head or face to be captured. If a child could not be captured on video for any reason, I arranged for the child to be seated in such away to be excluded from the observation and video-recordings.

Given that one of my roles within the SM project was to facilitate the majority of the Year 6 PD, it was essential that the teachers understood my role as a researcher for my doctoral study. I explained to the teachers that I was interested in their knowledge development as they learned to use Scratch and teach the ScratchMaths curriculum materials. I reassured teachers that I was not making judgments of their practice or assessing why they had not included aspects of the PD I had delivered with their lesson. I explained that we would use the post-lesson interview as an opportunity to explore any issues which may have emerged within the lesson, and that I would be able to provide SM support if needed outside of the context of the interview if required. I explained that the data collected within the observations and interviews would not be reported back to the school or to the ScratchMaths project but would be used to develop an understanding of the evolution of teacher’s knowledge as they used the ScratchMaths curriculum materials to teach mathematical concepts within the case study research.
The storage of my data complied with the Data Protection Act 1998 (DPA 1998) and UCL’s Data Protection Policy. The audio and video recordings and related transcripts were kept on my password protected laptop, and in a secure area of the UCL one drive, connected to my UCL password protected user account.

### 3.8 Summary

In this chapter I have provided and justified the theoretical approach for the qualitative multiple-case study with augmenting teacher episodes methodology adopted by this research. I argued that it was necessary to collect rich data using multiple methods such as classroom observation, teacher interview and task-based interviews to provide multiple lenses on teachers’ mathematical pedagogical technology knowledge as they learn to program and explore mathematical ideas using the ScratchMaths curriculum. Next, I detailed the sampling criteria and introduced the participants of the study providing a detail of the different backgrounds and experiences of teaching computing and teaching ScratchMaths for the three case study teachers and the of the additional augmenting teacher episodes. I discussed how I operationalised my MPTK framework for teaching mathematics through programming by characterizing lesson features that might be observable in a lesson which adhered to the original design of ScratchMaths by using 5E pedagogical framework. I detailed and justified the methods for data collection and provided the analytical process including the rationale for using and adapting the TRU framework to provide additional characterizations of what took place inside a classroom. Lastly, I discussed how I addressed any ethical issues which could present themselves in the research.
Chapter 4  The ScratchMaths curriculum and microworlds

4.1  Introduction

This mini chapter discusses the ScratchMaths curriculum and provides greater detail of the two ScratchMaths microworlds, Module 4 – Building with numbers – *Place value* microworld, and Module 5 – Exploring mathematical relationships – *Fireworks* microworld. At the end of the chapter, I provide a brief overview of the case study structure to signpost the presentation of the data which follows.

Note: Appendix I includes an overview and glossary of Scratch and Scratch terminology used throughout this thesis.

4.2  ScratchMaths curriculum

The microworlds used within this study are taken from the ScratchMaths (SM) curriculum (UCL ScratchMaths, 2017). SM is a two-year computing and mathematics-based curriculum for Key Stage 2 pupils (Years 5 and 6). The overarching aim of the curriculum is to enable learners to engage with and explore important mathematics ideas through learning to program using the free online programming environment Scratch. The curriculum is organised into six discrete modules. Three in Year 5 have a computing focus, and the mathematical ideas are implicit, whilst the three in Year 6 make the mathematics explicit and use the computational ideas developed in Year 5.
The microworlds in focus within this study are taken from Module 4: Building with numbers and Module 5: Exploring Mathematical relationships. An overview of the mathematics and computing concepts for both modules are shown in the Table 4.1 below.

<table>
<thead>
<tr>
<th>Module</th>
<th>Computing concepts</th>
<th>Mathematics concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>4: Building with numbers</td>
<td>Uses broadcasting to build models with synchronised parallel behaviours in several mathematical contexts.</td>
<td>Place value models as well as conversions of length, weight and time (including connections between analogue and digital representations)</td>
</tr>
<tr>
<td>• Building a place value model up to four places</td>
<td>Place value microworld.</td>
<td></td>
</tr>
<tr>
<td>• Converting place value model to represent time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Exploring the place value model for conversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5: Exploring Mathematical Relationships</td>
<td>Introduces the concept of variable. The effect of the order of commands in an algorithm is explored through the need to store multiple values (e.g., the number of sides and length of side to be drawn).</td>
<td>The invariant relationship between 360° total turn, number of sides and angle of turn is explored through drawing polygons.</td>
</tr>
<tr>
<td>• Drawing random polygon night sky scenes</td>
<td>Fireworks microworld.</td>
<td>Links to different types of mathematical relationship including proportionality and ratio in the context of drawing rectangles.</td>
</tr>
<tr>
<td>• Drawing mathematically similar rectangles</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 ScratchMaths overview of modules containing mathematics and computing concepts

The SM intervention also included professional development (PD), two days per year, both underpinned by an explicit “framework for action” (DiSessa and Cobb, 2004) known as the 5E pedagogical framework (see section 2.3.6). The teachers were encouraged to complete gap task activities, a task set at the end of a PD day to be completed by the teacher before the next PD session which took place several weeks later. The purpose of the gap tasks was for the teacher to continue to explore and develop their confidence with using Scratch alongside the computational and mathematical ideas introduced within each PD day as preparation for the second PD day.

4.2.1 Module 4 – Building with numbers - Place value microworld

The importance of counting has been emphasised by a number of researchers (Callahan and Clements, 1984; Fuson, 1990a; Kamii, 1986; Steffe and Cobb, 1988) as the basis for acquiring an understanding of unit and multiunit numbers. In particular, the progression of moving from using ones as the counting unit, to using ten, followed by using a tens-one counting strategy. Despite the wealth of research into children’s understanding of place value, poor success rates
continue in tasks involving relations between units (Houdement and Tempier, 2019). The ScratchMaths place value microworld is a programmatic representation of a place value mathematical system. The microworld was developed to behave analogously to a mechanical counter such as a bike odometer. A mechanical counter consists of a series of disks mounted on an axle with the digits zero through 9 marked along the edge. Each disk has a protrusion that after a full revolution, moves (nudging) the next disk to the left one increment. The right most disk moves after an event such as the rotation of a bicycle wheel. However, in ScratchMaths the model is built using sprites which ‘wear’ digit costumes to represent each place and utilizes the broadcast and receive functions of Scratch to replicate the mechanical interaction nudging between disks.

The SM Place value microworld consists of a ones sprite designed to have ten costumes available to wear and represent the digits 0 to 9. The Scratch block next costume when attached to the sprite simulates counting as it has the effect of iterating through the costumes each time the sprite is clicked (Figure 4.1).

![Figure 4.1 Ones sprite, costumes and next costume block to simulate counting by 1](image)

A tens sprite can be created in the microworld by duplicating the ones sprite and renaming it. The new sprite will be identical except for the sprite name and the sprite’s position on the screen. These two sprite attributes are equivalent to corresponding properties of the place value
system. Grasping the relationship between adjacent columns in a place value model, and what happens when a multidigit number ending in 9 is increased by 1, the point at which this relationship is made explicit, is critical to forming a concept of place value in base ten (Fuson, 1990b; Thompson, 2000). In mathematics this process is described as carrying, exchanging, or trading. The conceptual abstraction required to construct the relationship between the groups of objects represented by each digit makes this process particularly difficult (Kamii, 1982).

Programming the place value relationship requires a mechanism for sprites to interact using the broadcast and when I receive () blocks. After the ones sprite has increased by one from nine to zero, a message “add ten” will be broadcast by the ones sprite. The tens sprite will be listening for the specific “add ten” message, and when it hears the message, the block will react and change the costume of the sprite to the next costume as shown in Figure 4.2. The effect of changing the tens costume to the next costume, is equivalent to adding on ten i.e., performing the carry/exchange of ten.

![Figure 4.2 Script for Tens sprite to listen for the add ten message](image)

The costume worn by the ones sprite and displayed on the screen is a representation of a digit, a specific organisation of pixels. Scratch is not concerned with the detail of graphical costume, however Scratch can report on the costume number of the currently worn costume using the costume # reporter block. To program the exchange/carry of ten Scratch must operate on the costume # reporter block to check which digit is currently displayed using the if () then structure (Figure 4.3).
The implication of whether to place the digit zero costume at the start of the costume list or at the end of the costume list now becomes apparent. The value of the costume # reporter block for the digit zero costume will be either 1 or 10 depending on its placement. Mathematically, the digit 0 is dual purpose in a place value system, indicating the quantity (0) of a base ten unit as well as holding the place when there are 0 copies of a base ten unit. E.g., if there was no concept of zero in place value, the number 302, would be represented as 3 2, figuratively identical to the number 32. The digit 0 is therefore ordered first as in the whole numbers. However, for the Scratch place value representation, placing the digit zero costume at the start of the costume list has a somewhat confusing side effect on the costume #. Table 4.2 illustrates the two approaches to ordering the costumes.
As can be seen from the table when the digit zero costume is ordered first in the costume list there is a mismatch between the costume displayed on the screen and the costume # reported. I.e., the digit zero costume has a costume # which Scratch reports as 1. The script to program the exchange/carry would become Figure 4.4.

This script therefore has the potential to be confusing to the learner due to the mismatch between the digit displayed and the costume # operated upon. Additionally, it the learner has
not understood that the costume # reporter block is an index of the costume list, and not what is displayed on the digit, they may program if costume # = 0. This is an incorrect syntax and since the programming environment does not provide any feedback on the incorrect syntax, the learner would find it challenging to debug why their program is not working.

However, if the digit zero costume is ordered last in the costume list, the digit costumes between one and nine match their reported costume#. The script for the carry/exchange would become Figure 4.5.

![Figure 4.5 Script to check if costume # = 10 to represent the place value carry/exchange mechanism](image)

The use of 10 makes explicit the carrying/exchange at 10 and aligns with the pencil and paper process of multi digit addition. Thus providing an opportunity for the children to construct the algorithm for themselves at some level (Kamii and Dominick, 1998). However, a mismatch between the digit displayed and the costume # remains. The positional nature of the places can then be built into the model using (Figure 4.6). Designing the place value model in this way, the sprite becomes an object to think with. The ones sprite broadcasts a message to “add ten” when the costume changes beyond digit 9, and the tens sprite listens for the message and change its costume to the next one in the list, simulating carrying or exchanging. The power of this programmed place value model is strengthened when generalised beyond two digits, both in the programmatic representation and the instantiation of the mathematical idea, since the programmatic structure for each successive digit is the same as the previous digit.
The ScratchMaths materials also contained opportunities for children to engage with “half-baked microworlds” (Kynigos, 2007) where they were asked to make changes to a provided Scratch project and fix or adapt it. Kynigos defines a half-baked microworld as specially designed to incorporate an interesting idea, but it is faulty or incomplete. An example of this approach was the design of The Conversion Game. The point of the activity is for the children to deconstruct the game and make changes and construct a new artefact to explore place value in other contexts such as time, weight and money.

4.2.2 Module 5 – Exploring mathematical relationships - *Fireworks* microworld

The SM *Fireworks* microworld consists of a single sprite, a beetle viewed from above. The beetle can be dragged to a new position on the stage using the mouse or can be moved to a specific position on the cartesian plane by programming with the go to \( x : y \) block. The beetle can be controlled with the move and turn blocks and will draw if its pen is down as it moves around the stage. In ScratchMaths this is known as Beetle Geometry and is analogous to
Papert’s (1980) Turtle Geometry in Logo, but exchanges the notion of a turtle to that of a beetle. The beetle sprite (Figure 4.7) was selected to retain the properties of the on screen Logo turtle, e.g., it is clear which direction the beetle is pointing, unlike the default cat sprite which is viewed from a side perspective. Consequently, acting and playing beetle retains Papert’s ideas of body syntonicity where the learner knows understands the geometry of their own body and can use these ideas to think like a beetle as they move and turn, and then transfer these to the programming language.

Figure 4.7 Fireworks microworld beetle sprite viewed from above (analogous to Papert's turtle).

A regular polygon can be drawn with the beetle using a simple script. Figure 4.8 below shows exemplar beetle scripts to draw a square, a regular pentagon, and a regular hexagon.

Figure 4.8 Scratch script to draw a square, a pentagon and a hexagon

Common to each of the scripts is the repeat block, the move block and the turn block. Each of these blocks in Scratch is related to a geometric property in mathematics for a regular polygon:
• repeat – The number of sides in the regular polygon
• move – The side length of the regular polygon
• turn – The exterior angle of the regular polygon

In the ScratchMaths materials in Year 5, children explore the relationship between the values in the blocks and the output on the screen. The children discover that the beetle will turn through a total of 360 degrees as it draws a regular polygon (a variant of Papert’s (1980) Total Turtle Trip Theorem). Knowledge of this fact allows the children to build a generalised script (Figure 4.9) for any sided regular polygon. The script is built and explored in the Polygon fireworks investigation.

![Figure 4.9 General regular polygon script using answer block to represent number of sides](image)

The module contains several activities designed to develop the concept of a variable through increasingly complex contexts. For example, the first activity encourages children to explore the behaviour of the ask and answer blocks. The ask block will make the sprite say the question and show an input box at the bottom of the screen; the answer block will automatically update to the most recent input typed into the box. In Scratch, the ask block is known as a reporter block, one that reports a value. The properties of storing an input from the ask block and reporting the value in the answer block means that the answer block can be considered an instance of a variable. For example, in Figure 4.10 below, clicking the ask block causes the beetle to ask the question “What is your name?”. The value inputted is stored automatically in the answer block. In the second screenshot, the beetle reports the value of answer by using the say block and using the answer block in a sentence, by using the join block, which concatenates (joins together) the text of “Hello “ and “answer”.

The variable concept is developed further using the `answer` block in different activities which distance the `answer` block from where the `ask` block is located within the script. The rationale of this approach is motivated by how children’s understanding of variable develops (Küchemann, 1978) and the importance of developing the role of variable (Philipp, 1992) though the use of different contexts. Some examples of drawing a square which use and operate on the `answer` block are show in Table 4.3 below. For further details see the ScratchMaths curriculum materials (UCL ScratchMaths, 2017).
<table>
<thead>
<tr>
<th></th>
<th>Using variable to set something:</th>
<th><img src="image" alt="Code Snippet" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• set pen size to $\text{answer}$</td>
<td><a href="image">Code Snippet</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Using variable to control movement:</th>
<th><img src="image" alt="Code Snippet" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>• move $\text{answer}$ steps</td>
<td><a href="image">Code Snippet</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operating on variable:</th>
<th><img src="image" alt="Code Snippet" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>• change pen size by $\text{answer}$</td>
<td><a href="image">Code Snippet</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Operating on variable:</th>
<th><img src="image" alt="Code Snippet" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>• say join The perimeter is $4 \times \text{answer}$</td>
<td><a href="image">Code Snippet</a></td>
</tr>
</tbody>
</table>
The programming activities are carefully sequenced so that only one variable, the answer block, is needed. Script 5* asks a question to set pen size and then asks a second question to set the side length of the square. The answer to the first question is forgotten as it is overwritten by asking the second question. The order of the blocks is therefore important; the pen size must be set immediately after the question is asked. However, in Activity 5.1.3 shown in the PowerPoint slide in Figure 4.11 below, children are asked to consider what would happen to the value of answer if two ask blocks are connected.
The activity makes explicit that the **answer** block cannot be used to solve the problem, since two values cannot be stored inside the **answer** block simultaneously; the second **ask** block will replace the contents of **answer** (side length) with a new value (the number of sides). Unlike Script 5* above the **answer** block is distanced from the **ask** block. However, the problem can be solved by using two user defined variables. This approach provides opportunities to engage with the assignment of variables, a key misconception found in novice programmers (Doukakis, Grigoriadou and Tsaganou, 2007; Sorva, 2012). The variables are first created by the user, and then **set to** the **answer** block, which stores the answer to a question. Figure 4.12 shows the script for a general polygon using two variables, **side length** and **number of sides** set as responses to the two questions. Creation of the algorithm for creating a general polygon of any number of sides, and any side length, requires an understanding of relations between variables, knowing how input relates to output, and in what order operations have to be done and exemplifies algebraic thinking. The two variables can be operated upon to display the perimeter of the regular polygon e.g., **say number of sides * side length**.
Figure 4.12 Script to draw a regular polygon which asks for two user inputs and stores them in two variables:
number of sides and side length

The investigation builds iteratively to the outcome of a firework display of regular polygons over a skyscraper towers of squares in the London skyline as shown in Figure 4.13 below.

Figure 4.13 ScratchMaths polygon fireworks in the night sky Scratch stage output

4.2.2.1 Altering polygons: Activity 5.2.1

Another use of the microworld is Activity 5.2.1 (see Figure 4.14 below) which explores using simple algorithms to create sequences of squares patterns. The investigation starts by exploring three different blocks:
The ScratchMaths curriculum and microworlds

- **Set side length to** 30;
- **Change side length by** 10;
- **Square.**

Each of the blocks can be explained relatively easily since their literal names describe what the beetle will do i.e., the *set* side length to 30 block will *set the side length* variable to 30, the *square* block when clicked draws a square. However, when the blocks are connected, and run repeatedly, the output on the screen is unexpected since a pattern is drawn. It can be cognitively challenging to explain the output *and* the value of the variable used each time the square algorithm is run.

![Module 5: Investigation 2 Activity 5.2.1 - Sequence of Squares](image)

Figure 4.14 SM Activity 5.2.1: Sequence of Squares

For example, if the blocks are arranged as shown in Figure 4.15 below and run in this sequence: First click *set side length to* 50 and then then click the *square/change side length by* 10 pair four more times, the beetle will draw the square pattern shown on the left hand side of the screen.
An exemplar approach to *Explain* and reason what is happening to the variable and to the beetle as each block as the algorithm is processed step by step is shown in Table 4.4. This type of mathematical thinking and reasoning is therefore an expected outcome of the task and provides an opportunity for the teacher to develop children’s reasoning skills through their use of a step by step approach.

<table>
<thead>
<tr>
<th>Step</th>
<th>Value of side length variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><strong>Set side length to</strong> 50 steps</td>
<td>The variable is set to 50.</td>
</tr>
<tr>
<td>2.</td>
<td>Draw a <strong>square</strong> using side length variable</td>
<td>50 Beetle moves 50 steps to draw a square. Beetle starts and stops in the same place.</td>
</tr>
<tr>
<td>3.</td>
<td><strong>Change side length by</strong> 10</td>
<td>60 The variable increases by 10, to 60.</td>
</tr>
<tr>
<td>4.</td>
<td>Draw a <strong>square</strong> using side length variable</td>
<td>60 Beetle moves 60 steps to draw a larger square over the top of part of the first square drawn.</td>
</tr>
<tr>
<td>5.</td>
<td><strong>Change side length by</strong> 10</td>
<td>70 The variable increases by 10, to 70.</td>
</tr>
<tr>
<td>6.</td>
<td>Draw a <strong>square</strong> using side length</td>
<td>70 Beetle moves 70 steps to draw a larger square.</td>
</tr>
<tr>
<td>7.</td>
<td><strong>Change side length by</strong> 10</td>
<td>80 The variable increases by 10, to 80.</td>
</tr>
<tr>
<td>8.</td>
<td>Draw a <strong>square</strong> using side length</td>
<td>80 Beetle moves 80 steps to draw a larger square.</td>
</tr>
<tr>
<td>9.</td>
<td>… repeat steps 7 and 8 one more time.</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.4 Reasoning steps to produce a sequence of squares

Note: variable concepts are also *implicitly* used within Module 4 in expressions such as `if costume # = 10` for the SM place value model. The value of the `costume #` reporter block can only be changed through the `switch costume to` and `next costume` blocks. Therefore, the `costume #` block is a type of `variable` within the Scratch environment since it shares properties with user defined variable, and with the `answer` block. E.g., both variable and `answer` block store information that can be changed and both blocks can be used in mathematical expressions and operated upon. This implicit use of variable in Module 4 was part of the curriculum design to lay the foundations to user defined variables developed in the Polygon Fireworks investigation in Module 5.

### 4.2.3 Teacher support materials

Central to the SM intervention is the development of the teachers’ knowledge through use of SM teacher support materials and corresponding PD. The PowerPoint slides for each investigation contain pupil activity instructions and teacher discussion questions. The PowerPoint slides can be used by the teacher to structure the lesson, as well as an aide memoire to key ideas. Posters were created which can be displayed in the teacher’s classroom to support the development of key Scratch vocabulary. The *ScratchMaths teacher guide* (see Figure 4.16 and Figure 4.17) provide i) step by step instructions for each of the activities, ii) connections to content covered in Year 5 ScratchMaths and iii) additional support through specific Scratch steps to support the development of teachers’ knowledge of the Scratch interface alongside their own instrumental genesis of the tool.

I contributed to the development of the teacher support materials and had a specific ownership to develop the content shown in the right hand column titled *Mathematical connections*. The intent of mathematical connections of the teacher guide was to demonstrate how mathematics can be explicitly connected (*bridged*) to the computational activity and content shown in the left-hand column. The mathematical connections included additional representations, mathematical definitions as well as suggested teacher discussion questions.
Figure 4.16 ScratchMaths exemplar teacher guide for activity instructions from Module 5 – Exploring Mathematical Relationships - Investigation 1 – Activity 5.1.1: Ask and Answer (UCL ScratchMaths, 2017)
Figure 4.17 ScratchMaths exemplar teacher guide for additional support from Module 5 – Exploring Mathematical Relationships - Investigation 1 – Activity 5.1.1: Ask and Answer (UCL ScratchMaths, 2017)
4.3 Case study structure and emergent themes

In the next three chapters I present the case studies of Mary, Rina and Sally. Each case study includes description and analysis of data from lesson observations and interviews. The case studies are structured into three sections i) Window on mathematical knowledge for teaching, ii) How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum, and iii) Teacher pedagogic practices. In section 3.6 of Chapter 3 I outlined the analytical approach I followed for data analysis. Point vi) of the approach stated that I coded the observational data through the lens of the Mathematical / Computational Knowledge for Teaching component of the theoretical model. I did this by re-reading each observation looking specifically for expressions of mathematics, or mathematical pedagogy whether in the teacher’s language or in their use of Scratch or other representative forms. Two themes emerged during initial coding and constant comparative strategies (Strauss & Corbin, 1990) 1) Thinking like a sprite and 2) Thinking with your maths head on. The codes related to the themes, their properties, and examples of classroom dialogue are presented in Table 4.5.

<table>
<thead>
<tr>
<th>Code</th>
<th>Properties of code</th>
<th>Example classroom dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theme 1: Thinking like a sprite in terms of…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Sprite thinking place value)</td>
<td>Explains connection between physical digit card flip book and costumes worn by sprite as representing a specific place value. Explains next costume for sprite is the same as adding one in maths. Explains broadcast and when I receive Sprite behaviours represent carrying and exchanging ten.</td>
<td>“MW: Good. So, the ten seconds digit is going from zero, 1, 2, 3, 4, 5 because we are dealing with time, it doesn’t have ten costumes [like the flip book].” [DMW6(Obsv)] “RE: When we click the ones sprite, it goes up to costume number 9, then tens sprite it adds 1.” [ERE1(Obsv)] “MW: Oh very good, it is broadcasting, what is he actually telling the tens to do? P: Add another 10.” [DMW1(Obsv)]</td>
</tr>
<tr>
<td>(Sprite thinking variable)</td>
<td>Uses the sprite’s ask and answer behaviours to explain about setting and storing values. Uses answer or variable blocks to represent generalised number.</td>
<td>“MW: It goes into the answer block, so that’s like a little storage box, that will store whatever you write in for your answer.” [DMW8(Obsv)] “CC: So, it’s going to move whatever you put for the length of the sides, it’s going to move that many steps.” [ICCI(Obsv)]</td>
</tr>
<tr>
<td>(Sprite thinking angle)</td>
<td>Uses a cut-out beetle to demonstrate angle turned.</td>
<td>“SE: So if, we had the beetle, and we were to do the 4 that we've got a movement of, we would do,</td>
</tr>
</tbody>
</table>


The ScratchMaths curriculum and microworlds

Uses role play as beetle to demonstrate movement and angle turned. “Playing beetle”.

“Playing beetle”. we’d start here and we would go, 1, 2, 3, 4. Then what would we do?
P: Turn right 90.” [ESE4(Osv)]

“MW: Armin is going to draw with his feet an equilateral triangle. Who is going to give him the instructions? Make it small enough so that he doesn’t end up in any furniture.” [DMW7(Osv)]

**Theme 2: Thinking with your maths head on**

**Thinking maths head**

Reads and interprets scripts step by step.

Bridges between maths and computer contexts and makes context explicit.

Relates mathematical step to equivalent step in programming.

Uses programming unplugged activities to create output using pen and paper.

“MS: So if I attach this to this, I have a very simple script for making a square whatever size I want it. Set side length, and then square. Will that make a square when I press it?” [BMS2(Osv)]

“MW: Yes, it does flip a number, but in Scratch, what’s that doing?
P: Changing costume.” [DMW1(Osv)]

“KW: Let’s just take our Scratch heads off for a minute and put our Maths heads on. If this was a clock, what would this number represent?” [CKW1(Osv)]

“RE: As she was drawing she has written her numbers next to each of the sides to remind her which ones should be longer and which ones should be shorter. She has thought about what has happening, basically talking to herself, like she is a computer.” [ERE6(Osv)]

Table 4.5 Classification of codes and properties of codes to exemplify the two themes related to teachers use of similes which emerged from analysis of lesson observations

The two themes are used as subsections of each case study within the *How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum* sub-sections, Within these sub-section the themes are exemplified using expanded classroom dialogue from the lesson observations. The relationship of each research questions to aspects of the MPTK framework, and to the structure of the case study is shown in Table 4.6 below.

<table>
<thead>
<tr>
<th>Research question</th>
<th>Aspect of the MPTK framework</th>
<th>Structure of case study (x.x represents Chapter and section)</th>
</tr>
</thead>
</table>
| (RQ1) How does primary teachers’ knowledge and mathematical thinking of selected mathematical concepts shape their engagement with teaching microworlds around these concepts, and reciprocally. | Mathematical Content Knowledge
Computational Content Knowledge
Pedagogical Knowledge | x.2 Window on mathematical knowledge for teaching
• x.2.1 Window on place value
• x.2.2 Window on variable
• x.2.3 Window on angle |
### 4.3 Case study structure and emergent themes

<table>
<thead>
<tr>
<th>Question</th>
<th>Mathematical/Computational Knowledge for Teaching</th>
<th>x.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum</th>
</tr>
</thead>
<tbody>
<tr>
<td>how is their knowledge and thinking shaped by this engagement?</td>
<td></td>
<td>x.3.1 Thinking with your maths head on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x.3.2 Thinking like a sprite</td>
</tr>
<tr>
<td>(RQ2) How do primary teachers use the digital resources of the ScratchMaths curriculum in their teaching to orchestrate children’s learning?</td>
<td>Technology instrumental genesis Mathematical/Computational Knowledge for Teaching</td>
<td>x.4 Teacher pedagogical practices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x.4.1 Whole class orchestrations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x.4.x Orchestration exemplification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x.4.x The TRU framework</td>
</tr>
</tbody>
</table>

Table 4.6 Mapping of research questions to aspect of the MPTK framework and case study section
Chapter 5  Case study 1 – Mary (MW)

5.1  Introduction

In this chapter, I present a description and analysis of selected episodes from Mary’s lesson observations of teaching with the SM Place value and Fireworks microworlds and her responses to the task-based interviews; both of which open a window onto her mathematical pedagogical technology knowledge. The case is structured into three sections i) Windows on mathematical knowledge for teaching, ii) Technology shaping and shaped by the mathematics, and iii) Teacher pedagogic practices. The relationship of the three sections to the research questions and mapped on to the conceptual framework is shown in Table 4.6 in section 4.3.

5.2  Window on mathematical knowledge for teaching

5.2.1  Window on place value

Mary’s responses to the place value mathematical knowledge for teaching test items (see Table 3.2 on page 76) indicate that she was able to identify the child’s place value error and was able to give a correct solution to each problem. For example, in question 2, which explores the teacher’s understanding of two digit column-addition, Mary’s brief response of “that’s a silly mistake, … they are just thinking that you carry ten every time, whereas you can carry twenty, thirty etc.” [DMW(PVT-Q2)] suggests that she is aware of the error and can explain some of the child’s thinking, but not necessarily the cause of this misconception. This type of child misconception commonly develops if children have only experienced adding two two-digit numbers together since ten is the maximum that can be carried. Mary’s response to the following two place value questions also identified the mistake and provided a correct solution but did not suggest a reason for the misconception or comment on the child’s understanding of
place value. Her responses to both questions were that the children’s errors were “silly mistakes” or had “mixed things up” [DMW(PVT-Q1, Q3)]. Her lack of explanation could be a limitation of the question asked, so it is not possible to identify if Mary did not know the root of the misconception as presented in the test item, however her responses do suggest that she tended to explain the what of the mistakes rather than explaining the why.

**Place value structure and ‘tenness’**

Mary’s initial lesson [DMW2(Obsv)] exemplified how she focused on the role of ten within the *Place Value* microworld in her teaching (see Chapter 4 for more detail of the design of the microworld.) The key structural aspects of the model are the digit costumes available to the sprite and the conditional check if costume # = for the costume is currently displayed; both of these aspects are critical in defining the base ten of the model and give it its *tenness*. Mary asked the children what would happen to the model when one more was added to the ones sprite that displayed 9. In this lesson dialogue she explained the link between the computer representation of ten and the representation of ten in mathematics:

**MW:** When you change the costume number to after nine, what happens? What number is coming after nine?

**Child:** Zero.

**MW:** Is it zero?

**Child:** Ten.

**MW:** Why is it only showing a zero then?

**Child:** Because it is the ones digit, so it can’t have two numbers.

**MW:** So, we don’t have a tens?

**Pupil:** Column?

**MW:** Not column, you are right, in Maths it would be column. But what is it in this?

**Child:** Digit?

**MW:** What is that we have?

**Child:** Sprite.

**MW:** We do not have a tens sprite yet. We only have a ones. So, to show ten it has only got the zero, because where does the one show? For ten, it will show in the?

**Child:** …

**MW:** You’ve just said it.

**Child:** Tens sprite.
Commenting on the lesson dialogue, the child initially explained that the place or sprite can only hold the numbers 0 to 9, before needing an additional place or column. However, the child needed significant scaffolding from Mary through use of her direct questions to explain the need for an additional sprite, and how the tens sprite would represent the tens column in mathematics. To examine Mary’s own programming knowledge of the ScratchMaths place value model, she was asked to explain how it operated in the post-lesson interview. She explained:

The ones sprite is counting up, next costume each time, counting up all the ten costumes. And when it gets to ten it is broadcasting to the tens. To add ten to that. So, it switches costume and the tens sprite received it and does what it is told. [DMW3(Int)]

Mary’s explanation of the model captures the structural aspects of the model but contains an error since the sprites do not broadcast to other sprites, they broadcast a message and another sprite is programmed to listen for that message. This was a common misconception observed during the professional development sessions. However, the conditional check if costume # = 10 or as Mary explained “when it gets to ten” was explained correctly despite this being problematic for many teachers during the PD. As discussed in section 4.2.1, the use of ten can be counter-intuitive since when the costume displays a one, its index is one but when the costume displays a ten its index is zero. During pilot testing of the ScratchMaths materials, if costume # = 0 was found to be a common pupil error, yet one that did not emerge in Mary’s lesson and is likely a result of Mary making the connection between the sprites index and its value displayed explicit in the lesson. Mary was asked why she had used if costume # = 10 rather than if costume # = 0 in the post-lesson interview. Her response indicated an appreciation of the ‘tenness’ of the place value model.

It is not the number zero, it is the number ten. We would not broadcast for it to [check] to be zero, because zero comes before [in the costume list]. Zero shows the ones number in the ten and obviously the one is in the tens column.

Basically, zero and ten are different numbers, even though zero is in the ones column as zero. You are adding ten after nine, not zero, so they should know that. Even though, yes, they are looking at the number zero, it’s ten they are adding. [DMW1(Int)]

Mary’s explanations showed how zero and ten are represented in the microworld at the point when the ones digit (or place) changes from 9 to 0. She explained that there is an effect of
adding ten which must then be represented in the model by changing the *tens* sprite to its next costume. Her explanations suggest evidence to support that Mary is making a connection between the representations of place value in the ScratchMaths model and her mathematical knowledge of place value. To explore this claim further, Mary was asked to consider how the model could be adapted in the context of time. She explained:

> We’d put a colon to show minutes and seconds. And we obviously want it to stop at 60, so 60 seconds and 60 minutes. I suppose the costumes are going to be different. I haven’t looked at the time one yet. Obviously, it is going to count up to 60 and then broadcast, to change, well 59 and then on 60 it will change to a minute. [DMW(1)(Int)]

Her response demonstrated an understanding of the similarities between the place value and the time model as she had explained that the model will behave structurally in the same way, i.e., to reach a certain number and then broadcast. She was then asked how she would adapt the script from the provided ScratchMaths base ten place value model:

> So, when the costume \# = 60. Well, no, we won’t have 60 costumes… So, it will be like the tens. So, if we have the hours and the minutes, and now I’m thinking, should I be thinking this in thousands, hundreds tens and units. [DMW1(Int)]

Mary noticed an error in her explanation since she correctly explained that there would not be sixty costumes available to the sprite, although at that point of her explanation she had not detailed what the name of the sprite would be. Mary explained that the sprite would operate like tens but does not explain that to express time an equivalent place would be “ten seconds”, rather than minutes. However, her explanation supports the claim that she had made a connection between the ScratchMaths place value representation and the mathematical representation of place value and that she is beginning to make a generalised connection between the base ten place value model and time models.

### 5.2.2 Window on variable

Mary taught two lessons [DMW(3), DMW(4)] which used the *Fireworks* microworld to explore notions of variable using the *ask* and *answer* blocks. In the lesson she created variables by clicking the “Make a variable” button in Scratch and *set* and used *variables* in a generalised procedure for a polygon as shown in Figure 5.1. The polygon procedure represents the relationship between the angle turned, the number of sides, and the side length. In the post-lesson interview, she was asked to explain her understanding of variable.
Mary recalled encountering the term variable at her current school in an after school code club where a parent explained how to use variables in Scratch. She explained that other than that time, the next time she heard about variables was within the ScratchMaths PD. Mary was asked to explain what variable meant and found it challenging as illustrated by her explanation:

When it can have… oh I don’t know how to explain it. Can I google it? Is it when, can I answer in terms of… When there is more than one answer for something, or you can put different answers in? [DMW3(Int)]

I sensed her difficulty and asked to consider what she might say if a child had asked her to explain and to try to illustrate with an example. She continued:

Like when we were asking the question, and we had multiple answers. We needed to store the answer somewhere, and obviously we needed them to be different, so we put the answers into the variable. But, I don’t know. [DMW3(Int)]

Her response is correct, yet very computer centric and suggests that she is aware of the limitation of the answer block to store one value at a time. She has referred to the lesson she had just taught which used an activity to create a script to draw a regular polygon which required two variables to be set and stored simultaneously as shown in Figure 5.2.
Exploring Mary’s computational knowledge for variable, she was asked to consider if she had encountered other variables whilst using Scratch. It was anticipated that Mary might recognise that the `answer` block be considered a concrete instantiation of variable in Scratch (see section 4.2.2), since it can be set using `ask`, and shares the same computational properties within Scratch as other user defined variables. She was asked to consider if `answer` was a variable:

No, `answer` goes into setting the variable thing. Hang on, let me think of the code. So, answer can be a variable if you put it in, but you can also have it as random. [DMW3(Int)]

This response is somewhat confusing to interpret although I suggest that Mary is recalling when she had used the `answer` block to set a user defined variable as she had created in her script shown in Figure 5.2. As a new concept to Mary, it is likely that she is also considering the literal English meaning of variable, as she had said that it is something that can vary. However, considering variable only in the English context obscures the more nuanced definitions of variable in computational or mathematical contexts. E.g., that a variable can be something that that has one specific value and can be used and operated upon. It is also likely that Mary had not previously considered that `answer` might be an instance of variable since variables are easily created by clicking the “make a variable button”. Therefore, to examine Mary’s knowledge for variable she was asked to complete an interview based task using the Fireworks microworld (See section 3.4.2.2).
Mary was tasked to “Build a script which asks for the number of sides, and then draws a polygon of that number of sides.” Mary was quick to complete the task and talked through step by step what she did. Her final script is shown in the left hand side of Figure 5.3 below.

She demonstrated her script by running it: the script asked a question “how many sides”, she input a number 5, and the beetle drew then drew a regular pentagon. Although she had defined a variable for number of sides within the script, she had not actually used the variable. This suggests some evidence to support my earlier claim that Mary does not view the variable blocks and answer blocks as instances of the concept of variable, a critical aspect in developing an understanding of variable. However, the obsolete block was pointed out to Mary. Her response was to change the script to use the defined variable as shown in the right hand side of Figure 5.3. Mary as then asked to compare this solution with the original one and she explained that they both have the same output on the Scratch stage but that her first solution was inefficient as the variable is not used within the script. This response suggests that Mary has formed some connections in her knowledge for her development of the concept of variable which utilises both the answer block and user defined variables.
To examine Mary’s knowledge further she was then asked to operate on the variables (another critical aspect) and to develop her script so that it asks for the side length and then displays on the screen the perimeter of the shape. The task whose solution requires two variables to be multiplied together, would therefore provide another window onto her developing knowledge of variable. Her first attempt to the task is shown in Figure 5.4 below: Mary updated the script to use the two variables, side length and number of sides and correctly updated the move block. However, when displaying the perimeter, she had replicated the expression from the turn block. She ran the script, noticed the error and wanted to give up, but was prompted “what’s the perimeter of the shape?” she replied, “all the sides added together” and then excitedly stated that “it’s number of sides times by side length!”. The invention allowed her to update the operator to correctly express that the perimeter is found by multiplying the two variables together.

Figure 5.4 Mary's response to creating a script which asks for the side length, numbers of sides, sets side length and number of sides variables and then draws a regular polygon and displays its perimeter

Although Mary found it initially difficult to explain the concept of variable, examining Mary’s knowledge of variable using the task-based interview provided a window onto Mary’s evolving knowledge of variable. Within the interview Mary demonstrated setting variables and operating upon variable in the language of computer programming. She had no difficulty to express the mathematical formula for perimeter using two variables in the programming language. As a final note on Mary’s evolving variable knowledge, at the end of the task interview, Mary looked
up a definition of variable, and noticed that variable exists in multiple contexts, such as mathematics, science, and computing. She said “I’m able to teach all of that without knowing what you were trying to get of me. Is that weird?” Her response indicated that Mary was realising that her knowledge of variable might be enough to get her through the ScratchMaths lessons but needed to evolve further to allow her to make stronger connections between computing and mathematics contexts.

5.2.3 Window on angle

The first investigation in the Fireworks microworld has an overarching activity structure to create polygon fireworks in the sky using the idea of Beetle Geometry. In this episode from Mary’s observed teaching with the fireworks microworld [DMW3(Obsv)], she had shared the screen of a child who had tried to create an equilateral triangle. The child had used a turn of sixty degrees, (the internal angle of an equilateral triangle) but to the surprise of the child “it had made something else.” (Figure 5.5).

This surprising outcome is common when working with Scratch and was common finding in Logo research. I have taught a similar task multiple times over the past ten years to adults on undergraduate and postgraduate students in multiple disciplines, and it is always a powerful discussion why the angle turned is not sixty degrees. The type of angle in learners’ knowledge of angle in pencil and paper mathematics tends to be implicitly interior. For example, it is common to hear “the angles of a triangle are 180 degrees, or an equilateral triangle’s angles are 60 degrees”. However, when engaging with Logo or Scratch, the learner is confronted with
the turtle or here, the beetle as it turns through the exterior angle. Mary was prepared to engage her children in this discussion and guided them through an explanation. She started by drawing a representation of an equilateral triangle on the board, and said to the children, “Each angle inside is sixty degrees, because the sum of the angles inside the triangle is 180 degrees”. The children agreed with her and she asked them why the Beetle does not draw a triangle. One child suggested that the angle needs to be 120 degrees but was only able to explain that 360 divided by 3 was 120 rather than explaining the exterior angle. Mary adapted her diagram and extended the length of the side to mark the exterior angle as shown in Figure 5.6. She asked the children what the sum of the angles on a straight line were to help them calculate the exterior angle from the interior angle. She then walked through the sides of the shape using her pen to represent the beetle and asked the children how many degrees in total the beetle had turned. A child responds 360 and Mary summarised that “even though the internal angle of a triangle is 180 degrees, he turned all the external angles and ended up back where he started!”.

Figure 5.6 Mary's white board explaining the exterior angle of a triangle. She had marked 120 degree external angle and 60 degree interior angle. The right hand side shows the output of move 60 three times.

This classroom episode indicates that Mary had mathematical knowledge of interior and exterior angle and had engaged with how the Beetle turned in programming. She was able to explain the child’s error and through her bridging of the microworld to the pencil and paper representation.
5.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

In this section, I present exemplification of the *Thinking with your maths head on* and *Thinking like sprite* theme which emerged from the analysis across the teacher data. Within each theme, I will illustrate how the data characterises the theme and explain the relationship between the mathematical idea and the programmatic representation (see Table 4.5 on page 122 for classification of each theme).

5.3.1 Thinking with your maths head on

Investigation one using the *Place value* microworld has the overall objective to build and explore a working place value model of up to four places using which increases by one when the ones pace is clicked. The microworld contains four sprites each of which can wear digit costumes of 1, 2, 3, 4, 5, 6, 7, 8, 9 and 0. The sprites communicate with one another through the programming behaviours of *broadcast* and *when I receive*. In the observed lesson [DMW1(Obsv)] Mary first played the *Flip, Flip, Nudge, Nudge* game from the SM curriculum shown as in Figure 5.7 to start the children thinking about how to build a model using computer programming. Mary played the game with four children standing at the front of the class each holding a digit flip book containing the digits 0 to 9. When she rolled two dice, the ones place had to *flip* their flip book the number of times shown on the dice. When any child reached 9 and flipped to 0, they had to *nudge* their neighbour to the right who would then *flip* their flip book once. After playing the game a few times, Mary asked the children to *explain* what they had seen:
MW: What is happening when we are nudging?

MW: Tell me what is happening, Ruth?

Ruth: We are changing the unit, no, err… [she trails off].

MW interrupting: Harrison is the Ones, units, and he is counting up in ones.

MW: What happens and at what number does this happen?

Ruth: Every time we get to zero. Obviously when you count it goes like twenty, thirty.

MW: Why? Tell me why!

Ruth: Because err… [she trails off].

MW: Is it just because it is on zero? What does that actually mean in maths, Els?

Els: Does it mean… Is it because in normal counting it goes up a number every time, and this is asking Harrison to …

MW interrupting: What number is Harrison getting to?

Els: Nine and when he gets to nine, [I mean] if he gets past nine he has to nudge Matthew to flip one.

MW: Why, what number is past nine?

Els: Because it is ten.

MW: Ten, and when he gets to ten, he has to?

Children in chorus: Nudge!

MW: Why? What is the nudging doing?

Maddie: Broadcasting to the next…
5.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

**MW**: Oh very good, it is broadcasting, but what is he actually telling the tens to do?

**Child**: Flip?

**MW**: Yeah, what does that mean in maths?

**Child**: Add another ten.

**MW**: Add another ten, very good. So, when the ones get to ten, it is nudging the tens to add on another ten.

**MW**: Maddie correctly said that if we were doing that on Scratch it would be broadcasting from the ones sprite to the tens sprite, to tell the tens sprite to change numbers.

**MW**: What do we call that in Scratch, when it changes the number?

**Child**: Broadcast.

**MW**: We do broadcast the message, but?

**Child**: It’s receiving.

**MW**: It’s receiving and what is it doing when it receives add ten? What does it do?

**Child**: It flips a number.

**MW**: Yes, it does flip a number. But in Scratch, what is that doing?

**P shouts out**: Changing costume.

**MW**: Very good, Changing costume.

The dialogue illustrated how Mary gave the children opportunities to explain their thinking and to try to explain what they had observed happening with nudging and flipping in the context of the game and to use that explanation to bridge to Scratch. Notably, she doesn’t pick up on the child’s misconception of broadcasting to rather than broadcasting for all sprites to listen to as was also discussed in section 5.2.1 above. The children’s responses required additional scaffolding from Mary to connect the flip book representation with the representation in programming and the representation in mathematics. She bridged between the nudging action in the game, which tells the next place to flip their card, and the operation of adding 10 in the context of place value, when she stressed “what does that mean in maths”. Mary’s use of this phrase suggests she had developed a pedagogical strategy which exemplifies the theme of thinking with your maths heads on to bridge between the different contexts. She exemplified this strategy by asking the children to explain what is happening in the context of Scratch and connected the physical action of flipping a digit book to the Scratch programming behaviour of changing costume.
Mary also used this strategy at the end of the teaching episode when and asked the children to consider what is happening when the tens place nudges the hundreds place. As in the previous episode, her pedagogical approach was to bridge from the Scratch representation of nudging and changing costume to the mathematical representation of adding in place value. She encouraged the children to explain the mathematical context, however, the child made an error:

**MW**: Now before we move on to do this on the computer, when the tens gets to 0, what is it nudging the hundreds to do?

**Child**: Change costume in Scratch, flip a number.

**MW**: Give it to me in Maths, you are right in the game.

**Child**: Oh add another 10.

**MW**: Is it adding 10?

**Child**: 1?

**MW**: Think of place value.

**Child**: 100.

**MW**: It’s adding 100, so the hundreds have to add a hundred. And then Madison aka Hundreds is nudging the thousands, what is she telling the thousands to do?

**Child**: Change thousands

**MW**: Change thousands?

**Child**: Add thousands.

**MW**: Add thousands, very good.

In the dialogue Mary intercepted the child’s response when the child responded that nudging the hundreds means adding 10. It is likely that the child was repeating the same answer that had been correct in the previous class discussion for the tens place and thus ignoring the structure of the mathematical representation. Alternatively, the child’s second response could be considered correct since the digit had changed by one. It is uncertain what the child was thinking, however Mary’s decision to bridge to place value indicated that she had considered the importance of making connections between the Scratch programming representation and the representation in mathematics and had developed a pedagogical strategy to do so.

### 5.3.2 Thinking like a sprite

In this section I present Mary’s use of pedagogic strategies characterised by the theme *Thinking like a sprite* in three different contexts.
5.3.2.1 Thinking like a Digit sprite

Mary’s place value lessons [DMW1(Obsv) and DMW2(Obsv)] each began with the digit card flipbook activity *Flip, Flip, Nudge, Nudge*. Mary first played the game with the children and discussed how the game connects with programming a place value model in Scratch. For example, she opened the provided ScratchMaths starter project and asked the children to explore the *next costume* and *switch costume* blocks acting on the *ones* sprite. They were then asked to try activities shown in Figure 5.8.

![Image of Module 4: Investigation 1](image)

Figure 5.8 SM Activity 4.1.1: modify the script so that when the ones sprite is clicked it changes its value in different ways

The children explored the blocks and reported back that each time the sprite was clicked the sprite’s costume changed by one. They also explained that the *ones* sprite counted up to nine, changed to zero and started to count again as they continued to click. The children shared their different approaches to each of the questions from the activity. Some children had wanted to use the mathematics operator blocks but made little progress. To resolve this issue, Mary encouraged the children to imagine that they were the sprite and to *think like a sprite*, as they had played in the game. She reminded them that in Scratch the sprite can only use *next costume*
or **switch costume** to change what costume the sprite was wearing i.e., the digit displayed on the screen. The children had to think of *themselves* as the digit sprite since the microworld was designed so that the costumes that the sprite could wear were ordered in the same numerical order 1, 2, 4, 5, 6, 7, 8, 9, and 0 as in the flip book. Consequently, the Scratch behaviour of **next costume** has the effect of adding 1 to the digit displayed. This approach supported the children to only use the **next costume** blocks, or the **switch costume** blocks to perform the mathematical operation. E.g., to increase the ones sprite value by 7 can be programmed by using 7 **next costume** blocks, or using of the **repeat** block and **next costume**.

As the lesson developed, the pedagogical strategy of thinking like a sprite became more important. Mary asked the children to add a second sprite into the model (the tens) and use the **broadcast** and **when I receive** events to send messages between the sprites to mimic the behaviour of nudging as they had played in *Flip, Flip, Nudge, Nudge* game. Mathematically, these behaviours represent carrying or exchanging ten from the ones to the tens and programming this behaviour requires the use of a Scratch control block, **if () then** block. If its Boolean condition is true, the blocks inside will run, and the script will continue to be processed. The blocks are shown in Figure 5.9 below.

The conditional statement **if costume number # = 10** is needed when creating a base ten place value model. The use of ten in the statement makes explicit the carrying/exchange at ten, aligning with the pencil and paper process of multi digit addition. Mary encouraged the children to *think like a sprite* to connect together the multiple representations of i) the place value representation of the digit cards as played by children in in the starter activity, ii) the Scratch programming representation of the place value model and iii) the representation of the place value system in mathematics.
5.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

5.3.2.2 Thinking like a Beetle for variable

To support the children to think like a Beetle, Mary used activity 5.1.1 from the fireworks microworld see Figure 5.10 below.

![Figure 5.10 SM Activity 5.1.1: introducing ask and answer blocks](image)

In this activity, the children explore how the ask and answer blocks are connected to one another. They discover that whatever is typed into the box which appears on the stage when the ask block is clicked, is automatically stored inside the answer block. Mary asked the children to explore the ask and answer blocks and to explain what they notice. A child responded, “It keeps your answer.” and Mary replied “So, that’s where your answer is stored.” [DMW7(Obsv)]. To develop this further, she asked the children to program the beetle so that he says Hello to whatever is typed in. During the activity some children had programmed the Beetle without using with the answer block. For example, Leila’s screen is shown in Figure 5.11 below.
Although she had programmed the Beetle to say “Hello! Leila.”, as was required by the task, she had done this without using the answer block. Using the answer block is a critical aspect of developing a concept of variable, since the children had been asked to explore the relationship between what is typed in as response to the ask block and to explain what is displayed on the screen when the answer block is used. Discussing this observation with Mary in the post-lesson interview revealed that she had noticed the children’s error and rather than confront the error as a whole-class, she had told the child how to fix it. I asked her if she always worked in this way, and she explained:

I sometimes think I should let them [work for longer on their own], and some I would be happy to, but then it turns into carnage very quickly. I also do not want to lose half of them because they switch off if they think it is hard. A lot of them like the guidedness because they end up with something that works. [DMW6(Int)]

Given Mary’s developing knowledge of variable as discussed in section 5.2.2, telling the children what to do, rather discussing the error is perhaps a more comfortable pedagogical approach. However, Mary explained that in the next lesson she would concentrate more the on the ask/answer relationship before moving on to the next activity which used user-defined variables.

At the start of the next lesson [DMW8(Obsv)], Mary returned to the ask and answer task from the previous lesson. Her approach suggested an increased focus on what the beetle was thinking and doing. For example, she explained “Today we are going to focus more on the asking questions and getting the answers, and where the answers are stored.” Mary’s screen displayed the ask and answer blocks and the children were invited to explain what happens when they typed in their answer and where the answer goes. A child replied, similar to a response in the previous lesson that it goes into the answer block. However, Mary, elaborated what the beetle
was doing in her response: “It goes into the answer block, so that’s like a little storage box, that will store whatever you write in for your answer.” She then instructed the children to try out many questions and answers so that they developed the idea of the answer block as something that can be stored and operated on. She shared a question with the whole-class that had been asked during her walk around, a child had changed the question the beetle was asking to “What is your age?”, and the beetle responded with “Hello, 11”.

MW: Think about it. If the question is “What’s your age?”. Do we have to keep Hello in that box?

Children in chorus: No.

MW: What could we change it to that would make sense?

A child calls out: 11.

MW: No. Where would 11 be?

Child: In the answer.

MW: Good. So, we wouldn’t need to change the Hello to the answer, otherwise it would say 11, 11!

MW: Why would the beetle say 11 11?

Child: Because this is what he says first, then he says the answer.

MW: Good, so if we put 11 as the answer, he would end up saying 11 11. Or if we put in another age, he would end up saying 11, and then that other age.

MW: That is why I want you to start asking different questions and changing the way you adapt the script so that it always makes sense what the beetle says.

A child then suggested that you would change the beetle text hello, to “I am” so that the question and response of the beetle made sense.

5.3.2.3 Thinking like a Beetle for angle

The final context of thinking like a Beetle is situated in Mary’s lesson [DMW7(Obsv)]. The children had been asked to create different types of polygons shown in Figure 5.12 as preparation for the creating polygon fireworks in the night sky. Most of the class had drawn squares which used different pen sized and colours. However, one child had tried to draw an equilateral triangle and was perturbed by the output on the screen as he had programmed the beetle to turn sixty degrees which he thought was correct as shown in Figure 5.13.
Mary instructed the class to talk about the issue in pairs and were encouraged to try this out for themselves. One child suggested that the beetle needs to turn 120 degrees, because “the whole thing is 360 only, so you have to put in 120 to make the triangle instead of 60”. This is an accurate response to the question and indicated the child is considering the total turn of the beetle, however, when asked to explain the angle turned the child did not refer to the exterior angle. Mary then drew a diagram on the white board and explained that the beetle was turning through the exterior angle of the shape. To bridge the mathematics represented on the board to the thinking of the beetle, she asked one of the children to play beetle. She explained that “Armin is going to draw with his feet an equilateral triangle. Who is going to give him the instructions?”. With some guidance from Mary as the children gave instructions Armin
successfully walked out an equilateral triangle on the floor. To finish the episode Mary asked the children to confirm which angle Armin was turning when he played beetle, the children replied in chorus “the exterior”. Mary responded that they are correct, and repeated that the beetle turns through the exterior angle, and that the exterior angle can be calculated by dividing 360 by 3.

5.4 Teacher pedagogic practices

In this section, I identify and exemplify the kinds of whole-class activity types utilised by Mary as she taught SM activities using the Place value and Fireworks microworlds. I then go on to analyse the quality of Mary’s whole-class orchestrations and the adherence to the ScratchMaths curriculum through use of the adapted TRU framework for teaching mathematics through programming as discussed in section 3.6.4.

5.4.1 Whole-class orchestrations

Mary utilised most of the whole-class activity types in the eight observed lessons to orchestrate the use of Scratch programming and explore mathematical ideas. She used Scratch live in every lesson for at least 50% of the lesson time. Using Scratch live consisted of the following actions by Mary:

- building scripts in real time;
- running scripts, observing and discussing the screen output;
- debugging scripts, which combined editing the script, asking the children questions about the script and then running the script.

Exemplification of each of the types of activity across all lessons, as well as an indicative frequency are shown Table 5.1.
Table 5.1 Characterization and frequency of whole-class orchestration types as observed in Mary's lessons

<table>
<thead>
<tr>
<th>Whole class orchestration type</th>
<th>Characterization of whole-class orchestration</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical-Demo</td>
<td>Mary rarely performed a Scratch technical demonstration in the whole-class setting. She typically instructed the children to <em>Explore</em> a block directly so that they could discover how the block operated for themselves directed in the Scratch 5 E pedagogical framework. There were a few occasions where she noticed that the children were struggling with the Scratch interface during the children’s computer work. For example, she demonstrated how to drag the <em>answer</em> block in from the right of an operator block and also demonstrated how to make a new block.</td>
<td>Rarely</td>
</tr>
<tr>
<td>Discuss-the-screen</td>
<td><em>Discuss-the-screen</em> is Mary’s frequent whole-class technical orchestration used in each lesson. She asked the children mostly closed questions whilst using Scratch live rather than more open-ended questions where children were able to explain their thinking. For example, she asked the children where to find a specific block(s) in the Scratch palette or asked for an explanation of the behaviour of a particular block Although the questioning style did not encourage to expand or explain their given answer, Mary provided an explanation for their responses, effectively off-loading their thinking, rather than only correcting and providing encouragement. However, the children were able to explain some of their thinking more effectively in unplugged activities away from the computer as coded by Board-instruction.</td>
<td>Frequently</td>
</tr>
<tr>
<td>Explain-the-screen</td>
<td>This orchestration, by definition, connects the mathematical content with what is displayed on the screen. Mary used this sparingly and recognised the need to develop this further in her practice as she became more confident with the technology and knowing the mathematical links. [DMW2(Int), DMW5(Int)]</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Guide-and-explain</td>
<td>The <em>Guide-and-explain</em> orchestration was used most frequently when Mary explained how a particular Scratch script is expected to run, before she ran it. She adopted an approach working from the top of the script, and processing the meaning of each block, step by step from top to bottom. She used this orchestration to connect the output on the Scratch stage with the script. The Guide-and-explain orchestration for Mary is an example of her implicitly using <em>Envisage</em> of the Scratch 5 E pedagogical framework.</td>
<td>Frequently</td>
</tr>
<tr>
<td>Sherpa-at-work</td>
<td>Mary used this orchestration frequently during the eight observed lessons. She used the orchestration in two different ways i) asking a child to create a script as they developed through Mary’s whole-class discussions with the children and ii) asking a child to perform the numerical inputs for Mary as she explained a process. This latter approach allowed Mary to remain in front of the IWB (a large touch screen device) working live to drag out and reconfigure blocks and to draw attention to specific blocks which she would gesture to, but offloaded the data input, although possible with an IWB. Mary said that working in this way was a more efficient use of her time [Journ].</td>
<td>Frequently</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Board-instruction</td>
<td><em>Board instruction</em> was used frequently in all of Mary’s lessons teaching with the fireworks microworld but only once in the place value microworld lessons. Typically, this would be characterised by direct task instructions to the children, or direct action to perform within the Scratch environment without using Scratch herself, despite it being available. When Mary wrote on the IWB, for example, when showing the value of a variable changing as a script is processed, she did not significantly stress the relationship between what was happening in Scratch and the representation on the board, which would have indicated use of the <em>Link-screen-board</em> orchestration. Mary’s use of the unplugged ScratchMaths activities, which by design do not require the use of the computer, are also coded as <em>Board-instruction</em>.</td>
<td>Frequently</td>
</tr>
</tbody>
</table>
| Link-multi-modal-representation | In Mary’s place value lessons, she used a flip-book of digit cards to represent the Scratch representation of the sprite and the behaviour of changing costume. Four ring bound flip-books were used by the children during the *Flip, Flip, Nudge, Nudge* and *Nudge, Nudge, Get, Get* activities as well as a teaching resource to demonstrate the next costume behaviour of the sprite. (Note: a printable flip-book was included in the developed teaching materials.) Mary additionally repeated the activity at the start of each place value lesson as she felt that the children had enjoyed the activity and that the representation was particularly useful for the lower attainers who were not natural coders:  

> Because they can actually stop and think, ‘no that’s when I nudge’… because that is what they are actually thinking. And you can see when I was talking to them, they were actually going, ‘then you nudge, but what do we call that in coding, broadcast’ … so that brings it all to life! [DMW1(Int)]  

| Sometimes |
A first observation of the table is the high frequency of *Board-instruction* that occurred in the *Fireworks* microworld lessons. This can be explained as Mary had used the ScratchMaths curriculum PowerPoint presentations to structure her teaching. For example, in the slide shown in Figure 5.14, Mary read out the instructions and added additional explanation to clarify the steps on the slide. After she read out the instructions all children would be working on the steps displayed. Mary would occasionally stop the class when she had noticed something problematic a child encountered and then direct all children to how to resolve it. However, during this orchestration she did not demonstrate live or encourage the children to *explain* what they had noticed. Mary attributed her pedagogic behaviour due to the children’s relatively weak explanation skills. She explained that her own class were better at listening to her and were able to *explain* when she discussed a question that was not directly related to their own work. However, due to staffing challenges, Mary was also teaching two classes which were not her own. The children had an inconsistent experience the previous year with learning Scratch and so she explained that she took a more directed approach of telling them what to do, rather than providing them with greater opportunities to *explain* why there were doing it [DMW5(Int)].

Figure 5.14 SM Activity 5.1.3: consider if a script using two *ask* blocks and one *answer* block will draw a regular polygon of any number of sides and side length

Within the place value lessons, Mary used the PowerPoint presentations more sparingly and chose to teach the children using Scratch live more frequently, posing questions and giving her instructions verbally. An example of this pedagogic approach is illustrated by the *Guide-and-
explain orchestration below. Mary’s different use of the PowerPoint presentations in the variable and in the place value microworld seemed to reflect the different confidence in her developing MPTK for the corresponding content knowledge of variable and place value. I.e., she was more confident in the context of place value as she relied less on the structure of the PowerPoint presentations.

5.4.2 Guide-and-Explain orchestration

In the episode [DMW6 (Obsv)] Mary taught ScratchMaths activity 4.2.3 Countdown conundrum from the Place value microworld, see Figure 5.15 below. The objective of the activity is to explore how to build a countdown timer which uses multiple sprites to represent seconds and minutes and to engage with the concept of “borrowing” in mathematics. In the countdown model the one seconds sprite changes its costume to the previous costume to emulate time passing of one second. Mary had coded live the one seconds sprite so that when it reaches 0 it would change to 9 and then would then broadcast a message “Get ten seconds”. However, there was a bug in the script as the timer was not working as intended. When the digit changed from 9 to 0, nothing happened. The explanation of the bug is that the ten seconds sprite was not set to listen or receive the message; the two sprites were operating independently of each other. The following lesson dialogue starts after Mary had shown the children the one seconds sprite counting down and observed the bug when the countdown timer did not behave as they expected:

Figure 5.15 Mary pointing to the tens sprite on the interactive whiteboard. The children are building the countdown timer alongside Mary, she guides and explains each step.

MW: So that is counting down fine.
MW: Why is nothing happening in my ten seconds?

Child: It doesn’t know you’re broadcasting a message.

MW: Yes, it hasn’t received a message. We have to get it to receive.

MW: So let’s go to the ten seconds sprite, what do we need?

[She selects the ten seconds sprite.]

MW: You are really good at this, because you have done it in two projects already. How do we get a broadcast message to be received?

Child: Erm. I don’t know.

MW: You do know! You do! You’ve done it before!

MW: Just listen, how do I get them to receive the message?

Child: Oh, do you broadcast?

MW: I’ve told you! You knew. Well I’ve already broadcasted, I’ve sent the message. How do I get them to receive the message?

Child: When I receive?

MW: Excellent. I told you, you knew it!

MW: So, when I receive and what message am I receiving?

[She drags out a when I receive block].

All children callout: Get ten seconds!

MW: Get ten seconds.

MW: Now what order, again, am I going to do first, just like the last one?

Child: Erm, would you do the same as the last one?

MW: Yeah, you are going to do the broadcasting first and what message are we going to broadcast this time?

Child: Get one minute!

MW: Get one minute! So off you go, you are going to need the same blocks as you did before.

The dialogue exemplified the Guide-and-Explain orchestration as Mary’s questions and her live use of Scratch supported the development of how to debug the broken timer Scratch project. At each small step she asked the children to try to explain the expected Scratch behaviour and support their instrumental genesis of Scratch.

5.4.3 Link-multi-modal-representation orchestration

Mary used the Link-multi-modal-representation orchestration in both the place value and the fireworks microworld lessons. This orchestration can be observed when the teacher uses non-
digital representations such as flip books, or body movement to *bridge* between programmatic and mathematical ideas from within the technical environment but without accessing the technical environment. For example, the *Flip, Flip, Nudge, Nudge* and *Nudge, Nudge, Get, Get* activities.

In this episode [DMW6(Obsv)] Mary introduced the countdown timer place value model using an activity where children play as sprites to represent the places in the countdown timer. Three children had volunteered to represent the minutes, ten seconds and seconds place value as shown in Figure 5.16. The colon’s role was to check that what is happening in the timer is correct. Another child rolled two dice to represent the passing of seconds, the numbers called out, and the *one seconds* sprite is instructed to count down by that amount. If the *one seconds* sprite needs to flip beyond zero, they must immediately nudge the person to their right and “get 10 seconds”. Only then can they continue counting down. Each *sprite child* is therefore:

- observing what costume they are wearing as displayed on their digit book,
- waiting to flip their digit book when they are nudged,
- knows to nudge the sprite child to their right when they flip past 0.

![Children using place value flip books to think like a sprite as they engage with the countdown timer activity](image)

All the children in the class are encouraged to think through what is happening in front of them, and to count aloud and to indicate when to nudge. Before the dice was thrown Mary reminded the *one seconds* sprite how their nudge would connect to the Scratch programming language:

**MW:** When do you think you’re going to have to get ten seconds?
Child: When I go back to 9 again.

MW: So yeah, when you are on zero and if you have to keep going and have no more numbers, you are going to have to get from the ten seconds. You’ll have to broadcast a nudge, a get, and then before you can keep counting down. So, that’s the important difference [between the place value model built in the previous lesson].

The activity started with the sprite children set to 3, 2, and 5 representing 3 minutes and 25 seconds. Another child rolled two dice and said aloud seven, this prompted the class to count aloud at the same time as the one second child sprite flipped their digits. When the one seconds sprite had counted to 5, he nudged the ten seconds sprite, prompting her to flip her digit book as he continued to count up to seven. Mary then summarised what the sprite children had performed:

MW: “Connie you were on 2, [which is] twenty seconds, and he’s taken ten from you, so you’re on ten. So now we are on three, eighteen.”

The dice were thrown again, two fives, and the activity continued. The ten seconds sprite child changed to zero and she asked if she needed to get from the person next to them. Mary interrupted the game to discuss the question:

MW: So, Connie’s asked a question, does she have to get?

Child1: Yes!

Other children: No!

MW: Who says yes, who says when she gets to zero she has to nudge?

[Some hands go up, some are unsure.]

MW: Why? [Asking a child who said No.]

Child: Because you still got eight seconds left.

MW: Good, so we are still counting down, and if she nudged straight away, it would have gone to two, fifty but we would have skipped ten seconds. So, we still have to keep going down to zero seconds.

MW: When James is on zero and he needs more, he needs ten, then Connie is going to nudge.

The children were asked to consider the context of time since Connie wanted to change her ten seconds flip book from 0 to 9. Mary asked the children what number would come next since there are not ten costumes (digits in mathematics), available to the ten seconds sprite. The relationship between time in mathematics and the representation as sprite and costumes is reinforced when four different children perform the activity. By doing this Mary ensured that many children got to experience playing sprite [over the course of several lessons] and
supported the children with their developing understanding of connecting the programming representation of time to the mathematical context of time. Mary explained that in the mathematical context of time when counting down by one second from one minute, the next number would be fifty nine seconds. However, the child sprites had displayed ninety seconds, because the ten seconds child sprite’s flip books contained the digits 0 to 9. At a later point in the lesson, Mary made a connection back to this moment by asking the children how many costumes the ten seconds sprite was wearing. This interplay between playing sprite the context of time, the mathematical representation of time and programming representation of time in the microworld characterises the Link-Multi-modal-representation orchestration.

5.4.4 The TRU framework

In this section I report on the analysis of Mary’s observed lessons through use of the adapted TRU framework. As explained in Chapter 3 Methodology section 3.6.4. Each dimension is scored on a 5 point scale: 1, 1.5, 2, 2.5, 3 (see Appendix A for the full scoring rubric). Each 10-15 minute period of the lesson is scored and a weighted average for each dimension is then calculated. The decision to present results in Table 5.2 and as a chart in Figure 5.17 to one decimal place captures the nuance in each lesson’s scoring. Agency as defined by the TRU framework rubric (see Figure 3.1 on page 96) is the students’ willingness to engage in classroom conversations through the teachers' provision of opportunities for the students to make mathematical conjectures, explanations and arguments.

<table>
<thead>
<tr>
<th>Date</th>
<th>Microworld</th>
<th>Mathematics through programming</th>
<th>Cognitive demand</th>
<th>Equitable access to content</th>
<th>Agency, Ownership and Identity</th>
<th>Formative assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/12/2016</td>
<td>Place value</td>
<td>2.2</td>
<td>2</td>
<td>3</td>
<td>1.3</td>
<td>1.4</td>
</tr>
<tr>
<td>02/12/2016</td>
<td></td>
<td>2.1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>12/10/2017</td>
<td></td>
<td>2</td>
<td>2.1</td>
<td>3</td>
<td>1.4</td>
<td>1.3</td>
</tr>
<tr>
<td>02/02/2018</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.4</td>
<td>1.3</td>
</tr>
<tr>
<td>13/06/2017</td>
<td>Fireworks</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>16/06/2017</td>
<td></td>
<td>2</td>
<td>2.2</td>
<td>3</td>
<td>1.7</td>
<td>1.6</td>
</tr>
<tr>
<td>09/03/2018</td>
<td></td>
<td>2.2</td>
<td>2</td>
<td>3</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>23/03/2018</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>
A first observation is the consistency of high scoring across dimension 3, Equitable access to the content. Mary’s lessons were very well structured, she set clear behaviour expectations which were met by the children and she encouraged participation from as many children as is possible, rather than choosing to engage with those that always had their hand up. There appeared to be established participation structures which resulted in engagement with the lesson activities, and meaningful mathematical participation. For example, during children’s computer activity, she circulated the classroom checking on those students who she knew may struggle, as well as checking on the higher attainers to ensure they were sufficiently challenged. There was no evidence during the observations of children disengaged with the lesson activities.

**Dimension 1, Mathematics through programming**

In each lesson Mary adhered to the ScratchMaths curriculum and engaged with the 5E pedagogical framework. She provided opportunities for the children to *Explore* new blocks, for example in the variable lessons exploring the ask and answer pairing and encouraged the children to explore different questions. When a new starter project was used, she always asked the children to *Explore* and then *Explain* the set-up script. Mary encouraged the children to...
predict, an example of *Envisage*, the output on the screen of scripts during unplugged activities, or before a script was run for the first time. For example, the children were asked to predict (*Envisage*) the output of the script as shown in Figure 5.18 in pairs. The question created an explosion of noise in the classroom. Some pupils were then selected to *explain* their thinking following their paired discussion. Mary encouraged the child to step through the script and captured on the board the value of the side length as the script processed.

![Figure 5.18 Script to draw 5 squares of increasing size as the variable increases](image)

However, Mary did not consistently *bridge* to ideas in mathematics when there were opportunities to do so. For example, when introducing the children to use variable in Scratch in the different activities, she was unaware that she could have made a link between variables, and algebraic representations in mathematics. [DMW3(Int)]. Mary recognised this limitation in the post-lesson interviews and said that she needed to develop this further in her practice. She noted that she expected this would develop as became more confident with the technology and was able to identify further mathematical links. [DMW2(Int), DMW5(Int)]

From the scoring rubric (see Appendix A page A-1):

*Proficient: The teacher uses orchestrations and engages with the 5E pedagogical framework, but connections with mathematics (bridge) are limited. Students are given some, but limited opportunities to Explore, Explain, Envisage and Exchange when working with the computer.*

**Dimension 2, Cognitive demand**

The cognitive demand dimension represents the type of activities used by the teacher and how the teacher scaffolds and supports students in productive struggles as they build understanding and engage in mathematical processes. The ScratchMaths curriculum was designed to provide
rich activities opportunities for mathematical thinking to all students of all attainments, however it requires the teacher to help children make connections between computational and mathematical ideas. Mary consistently scored at least 2 (proficient) in this dimension indicating that she is using the ScratchMaths activities as intended, but most frequently tends to take a more prescriptive approach which can “scaffold away” the challenges. For example, Mary frequently worked with the children in a whole-class setting, by using Scratch live. This allowed her to keep the children together, rather than allowing the children to work more independently. She explained that when she had given them freedom to work independently for longer periods, it was more chaotic as she was unable to deal with the vast difference of attainment within the class.

From the scoring rubric, Proficient: Classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to "scaffold away" the challenges, removing opportunities for productive struggle.

**Dimension 4, Agency, Ownership and Identity / Dimension 5, Formative Assessment**

The analysis of the whole-class orchestrations suggested that Mary often provided little opportunity for the children to explain their own thinking and reasoning. For example, Mary frequently programmed live at the front of the classroom for the children to follow. She asked questions of the children at each step however the children often gave very short answers to Mary’s closed questions. The result is that the children were able to follow the programming steps simultaneously with Mary but when she sought a more reasoned explanation from a child, she often provided the explanation for them, rather than allowing the child an opportunity to develop their one word response. This approach is unsurprising, given Mary was teaching a curriculum for the first time, and was therefore less confident to allow the children more freedom. Mary explained that she often taught a Scratch lessons three times in a day, and by the third she felt much more confident and was able to provide more opportunities for the children to contribute to whole class discussions. In the post-lesson interview [DMW6(Int)] she reflected upon whether she directed them too frequently wanting confirmation that this was ok. I explained that she must use her professional judgement as in any lesson as she assessed the progress the children are making. Over the course of all the observed lessons, there is some
evidence of development in this dimension as Mary’s confidence grew and provided more opportunities for the children to explain their thinking.

From the scoring rubric:

Dimension 4: Mary’s classroom discourse lies somewhere between Basic: The teacher initiates conversations. Students’ speech turns are short (one sentence or less), and constrained by what the teachers says or does and Proficient: Students have a chance to explain some of their thinking, but the teacher is the primary driver of conversations and arbiter of correctness. In class discussions, student ideas are not explored or built upon.

Dimension 5: Mary’s use of formative assessment lies somewhere between Basic: Student reasoning is not actively surfaced or pursued. Teacher actions are limited to corrective feedback or encouragement and Proficient: The teacher refers to student thinking, but specific student’ ideas are not built on (when potentially valuable).

5.5 Summary

In this chapter, I have presented an analysis of Mary’s mathematical pedagogical technology knowledge (MPTK) organised as i) windows on mathematical knowledge, ii) mathematical ideas mediated and mediated by the ScratchMaths curriculum, and iii) teacher pedagogic practices. The analysis of Mary’s MPTK for place value suggests that she was able to solve and respond to common misconceptions of children engaging with pencil and paper calculations, but her explanations were limited to correcting the mistake rather than explaining why the misconception had occurred. Mary was able to make connections between the Scratch representation of the place value model and its mathematical structure including the broadcast behaviour to represent carry and the significance of ten within the model. Mary demonstrated that she was beginning to think of time as a place value system that operated with the same structure as place value in base ten but with different conditions such as place names and number of costumes or digits available for each place. There is some evidence to suggest that Mary’s knowledge of the concept of variable in computing is developing new connections to her knowledge of variable in mathematics after teaching with the Fireworks microworld. She
was able to use variable in programming in different contexts, such as expressing a perimeter using two variables operating on one another and was beginning to consider how variable in computing is related to variable in mathematics through her use of expressions.

The data illustrated three different ways Mary used the *Think Like A Sprite* simile, in the contexts of place value, variable, and angle, as an effective strategy to make connections between representations in Scratch and in mathematics. Mary’s main pedagogic practice was to either instruct the children without use of technology using the ScratchMaths PowerPoint presentations or to code live and simultaneously with the children, utilising the whole-class orchestrations *Discuss-the-screen* and *Sherpa-at work*. She used Scratch live for at least 50% of the whole-class activity time. She used and linked multiple representations: the white board, working in Scratch, the place value digit cards, and playing beetle. Mary developed a strategy “What does this mean in maths” as an effective way to bridge between the computational and the mathematical contexts.

Analysis of the observations using the adapted Teaching for Robust Understanding (TRU) framework suggested a proficient use of the ScratchMaths materials (*Cognitive Demand* dimension). Mary consistently used the provided materials, the PowerPoints, the teacher guide in each lesson, although she did not adapt the presentations in any way. In the *Mathematics through programming* dimension, Mary attended to some aspects of the 5E pedagogical framework, mostly utilising the *Explore* and *Explain* strategies, however connections between the programming and mathematics i.e., bridge were more limited. Mary encouraged children to explain their thinking (*Agency, Ownership and Identity* dimension), however children’s responses were often limited to one sentence or less. In these situations, Mary would respond to the children’s explanations by providing an explanation of what she thought the child was thinking rather than asking additional clarifying questions (*Formative assessment* dimension). Mary’s pedagogic approach (*Equitable access to content* dimension) ensured participation from all students through her use of well-established participation structures and routines.

Table 5.3 below provides a summary of the findings and their location within each section of the chapter where they are exemplified.
Table 5.3 Table to show summary of findings and location within the chapter for Mary’s case study

<table>
<thead>
<tr>
<th>Window on mathematical knowledge for teaching</th>
<th>5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Window on place value</strong></td>
<td></td>
</tr>
<tr>
<td>• Identified and corrected children’s errors in written place value tasks.</td>
<td>5.2.1</td>
</tr>
<tr>
<td>• Explained programmatic structure of Scratch place value model including broadcast &amp; receive.</td>
<td></td>
</tr>
<tr>
<td>• Recognised importance of the tenness of the place value model.</td>
<td></td>
</tr>
<tr>
<td>• Developed connections between Scratch place value model and Scratch stopwatch model.</td>
<td></td>
</tr>
<tr>
<td>• Beginning to think about time as an example of a different place value.</td>
<td></td>
</tr>
<tr>
<td>• Developed connections between place value models in Scratch and own mathematical knowledge.</td>
<td></td>
</tr>
<tr>
<td><strong>Window on variable</strong></td>
<td></td>
</tr>
<tr>
<td>• Computer centric explanation of variable.</td>
<td>5.2.2</td>
</tr>
<tr>
<td>• Used Scratch programming to help make sense of variable concept.</td>
<td></td>
</tr>
<tr>
<td>• Used expressions in Scratch to represent a mathematical formula e.g., perimeter.</td>
<td></td>
</tr>
<tr>
<td><strong>Window on angle</strong></td>
<td></td>
</tr>
<tr>
<td>• Moved between representations of geometric ideas: in programming, on screen and on paper.</td>
<td>5.2.3</td>
</tr>
<tr>
<td>• Developed connections between geometric structure in mathematics and in programming.</td>
<td></td>
</tr>
</tbody>
</table>

**How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum**

<table>
<thead>
<tr>
<th>Thinking with your maths head on</th>
<th>5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interpreted scripts step by step to encourage mathematical reasoning.</td>
<td></td>
</tr>
<tr>
<td>• Developed “What does this mean in Maths?” as an effective way to bridge between different representation such as nudging and adding 10 e.g., thinking in Scratch and thinking in Maths.</td>
<td>5.3.1</td>
</tr>
<tr>
<td>• Used unplugged activities to foster reasoned explanations and to envisage the output of programming.</td>
<td></td>
</tr>
<tr>
<td><strong>Thinking like a sprite</strong></td>
<td></td>
</tr>
<tr>
<td>• Used the sprite and its costumes as objects to think with.</td>
<td>5.3.2.1</td>
</tr>
<tr>
<td>• Used the beetle’s ask and answer as objects to think about variable with.</td>
<td>5.3.2.2</td>
</tr>
<tr>
<td>• Used the beetle as an object to think with when drawing polygons.</td>
<td>5.3.2.3</td>
</tr>
</tbody>
</table>

**Teacher pedagogic practices**

<table>
<thead>
<tr>
<th>Whole class orchestrations</th>
<th>5.4.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Used Scratch live and debugged live for at least 50% of the whole-class activity time.</td>
<td></td>
</tr>
<tr>
<td>• Frequently used Sherpa-at-work, Guide-and-explain, Discuss-the-screen, and Link-multi-modal-representation whole-class orchestrations.</td>
<td></td>
</tr>
<tr>
<td>• Frequently used some of the 5 E’s to make connections between Scratch and mathematics [D1].</td>
<td></td>
</tr>
<tr>
<td>• High level of engagement with SM curriculum [D2].</td>
<td></td>
</tr>
<tr>
<td>• Consistent classroom routines for children and teacher to engage with each other [D3].</td>
<td></td>
</tr>
<tr>
<td>• Teacher owned the classroom talk. High child participation, but low child agency, e.g. children’s thinking is not always surfaced. [D4, D5]</td>
<td></td>
</tr>
<tr>
<td><strong>The TRU framework</strong></td>
<td>5.4.2, 5.4.3</td>
</tr>
<tr>
<td>• The TRU framework</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The TRU framework</th>
<th>5.4.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Used Scratch live and debugged live for at least 50% of the whole-class activity time.</td>
<td></td>
</tr>
<tr>
<td>• Frequently used Sherpa-at-work, Guide-and-explain, Discuss-the-screen, and Link-multi-modal-representation whole-class orchestrations.</td>
<td></td>
</tr>
<tr>
<td>• Frequently used some of the 5 E’s to make connections between Scratch and mathematics [D1].</td>
<td></td>
</tr>
<tr>
<td>• High level of engagement with SM curriculum [D2].</td>
<td></td>
</tr>
<tr>
<td>• Consistent classroom routines for children and teacher to engage with each other [D3].</td>
<td></td>
</tr>
<tr>
<td>• Teacher owned the classroom talk. High child participation, but low child agency, e.g. children’s thinking is not always surfaced. [D4, D5]</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6  Case study 2 – Rina (RE)

6.1  Introduction

In this chapter, I present a description and analysis of selected episodes from Rina’s teaching with the SM Place value and Fireworks microworlds and her responses to the task-based interviews; both of which open a window onto her mathematical pedagogical technology knowledge. The case is structured into three sections i) Windows on mathematical knowledge for teaching, ii) Technology shaping and shaped by the mathematics, and iii) Teacher pedagogic practices. The relationship of the three sections to the research questions and mapped on to the conceptual framework is shown in of section 4.3 of Chapter 4.
6.2 Window on mathematical knowledge for teaching

6.2.1 Window on place value

Rina’s response to the first Place Value MKT question (see Chapter 3 - Table 3.2) indicated that was able to explain that the children had ignored the decimal places and that they were reading the numbers without using the decimal place. She explained that this type of error was usually not seen amongst her children, due to the drill and practice approach she takes with this aspect of place value. However, she explained that if the children did make this type of error, they would likely have made an attempt to line them up first. [ERE(PVT-Q1)]. Rina was quick to spot the children’s calculation errors in their column addition algorithms:

[Part i] They've not carried 20, they've carried ten. They've not carried the right number of tens not exchanged, that should be 22, but they’ve written 12.

[Part ii] That should be 21 but they've written 11.

[Part iii] That one they've carried the right tens, but not written the right units, that should be 15 but they've written 14. They've understood the method of sticking a ten up there, but they've not understood that they are actually exchanging two tens which is 20. This one, they have miscalculated. [ERE(PVT-Q2)]

Rina did not explain that the error of carrying ten could be a result of always experiencing adding two two-digit numbers together, since ten is always carried, although she did provide some insight into the children’s understanding of carrying ten. In Question 3, she correctly identified that the error is likely the child’s understanding of place value, rather than a mistake. E.g., the child had not recognised the 2 in 23 as worth two tens. However, she was quick to remark that the child could have been using place value counters which would make the child’s response correct since the child would still count out two counters and it would be important to ask the child to explain their meaning of 2 in 23.

Place value structure and extending the model

In a post-lesson interview [ERE1(Int)], Rina was asked to explain how the place value model in the microworld functioned. She confidently explained that the ones sprite had been programmed so that when it was clicked it would change to the next costume and elaborated how the sprites interacted with one another as evidenced by this narrative:
They've also programmed it separately to be ready to broadcast so when it gets to, from 9 to 0, or costume \[\text{number}\] whatever that is, it will broadcast whatever message it likes, and they've programmed this sprite to be listening out for the message which will tell it to change to next costume. [ERE1(Int)]

Rina’s response (unlike Mary) correctly explained that the Scratch broadcast and receive mechanism behaved like a ‘shout’ rather than broadcasting to a particular sprite. Thinking that the sprite broadcasts to, rather than a shout might be assumed after playing the Nudge, Nudge, Flip, Flip game, where the nudge metaphor is used to bridge to programming language. However, Rina’s explanation of the model demonstrated her understanding of representing the computational idea in this way. Rina also explained the generality of the place value model and described how she would extend the two digit sprite model to three digit sprites. Her explanation focused on the general features of the model, rather than a specific case:

If I had duplicated the units, I would have to delete a lot of the code in there because I wouldn't want it to do anything when they click it… Then I would edit the code in the tens to broadcast whenever it gets to whatever costume number. Then broadcast whatever message it is broadcasting. Then I would code my next one to be listening out for the broadcast. [ERE1(Int)]

Rina was asked to reflect on how she had used the SM place value model in her mathematics lessons or was planning to use the model in future lessons. She explained that using the same model to represent time as in the stopwatch activity is a good opportunity for pupils to think about other contexts and to “consider where they are exchanging and why [it is] necessary” [ERE1(Int)] rather than rote learning an algorithm. Rina explained that she found time a notoriously difficult topic to teach. She recounted a recent situation when a child had difficulty working on an examination question to calculate 35 minutes later than 4:30. She noted that she had observed the child trying to use column addition unsuccessfully. In the past, when she had encountered children working in this way she would say “You can’t do column addition, don’t do column addition” without providing a reason. However, felt that she would now be able to explain why as result of building the time place value model and explaining that there are different rules for the place value time model. “It kind of works in your head, cos you know you’re going to do that, and you can actually use column addition if you know the rules”. [ERE1(Int)] Rina is of referring to ‘a rule’ if you were adding 35 to 30, you calculate 65 and carry 1 (representing ten seconds) to the minutes. This is complex mental mathematics, but it demonstrates how she has connected place value with time. She thought that the Scratch model
would be interesting to use when she revisited time later in the year. These remarks suggest that Rina is beginning to think about the concept of time as an example of a (different) place value, a connection which may not have existed previously.

Rina was observed teaching a lesson based on building a stopwatch using the *Place value* microworld [ERE1(Obsv)]. In the lesson Rina had prepared her own Scratch project, a broken stopwatch for the children to debug. The project contained five bugs which Rina had created and embedded deliberately into the stopwatch; the task is explained in more detail in the *Spot-and-Show* orchestration exemplification in section 6.4.2 below. Rina’s attention to the main structure of the place value time model as built into her broken stopwatch, e.g., when to change digit for each sprite place in the stopwatch and the costumes available to the sprite, support the claim that Rina was developing connections between the different place value models as built in Scratch and her own knowledge of place value. Rina’s development and use of a “half-baked microworld” (Kynigos, 2007) was explained as a result of her deliberately encouraging an editing style in English lessons where she explained that it is “OK to make mistakes and to go back and change them.” [ERE2(Int)].

### 6.2.2 Window on variable

A variable is something that changes. If I think about it in terms of science, you control variables, and only change one. In an investigation for example ‘What conditions do plants need to grow?’, you only change one thing. So, if you were doing light levels, you want to make sure that all other variables stay the same: water, air and space, but you would vary the light. You would control the other variables you would keep them the same.” [ERE4(Int)]

Rina did not recall when she first encountered using variable in computer programming, although she did recall using variables in a Science context during her undergraduate degree and when she had edited HTML script. After teaching four lessons using the *Fireworks* microworld Rina was asked about her understanding of variable. The dialogue shown in Table 6.1 illustrates how Rina responded to the question [ERE4(Int)].
Table 6.1 Rina explaining her current understanding of the concept of variable.

Commenting on the table, Rina’s response to the questions provided evidence for the claim that that her understanding of variable in Scratch is connected to her understanding of a variable in the context of algebra in mathematics. She described representing variable as the \textit{side length} of the polygon in the polygon fireworks script and as \textit{number of floors} for a tower of squares. She also provided an example of operating on variable using multiplication in an expression; these are both explicit uses of variable through use of Scratch blocks. Rina was asked to consider if the \texttt{costume #} block, although not explicitly labelled a variable is an instance of the concept of variable (see section 4.2.2 for further detail). She responded:

\begin{quote}
Yes, because it is something that you can set, something that we can input. The road that I’m going down is that a variable is something that we control separate to the machine, something that the human does. [ERE4(Int)]
\end{quote}

However, she struggled to refine her definition:

\begin{quote}
In my head, I think in pictures, which is why I can’t articulate it. It is something that I am setting and it stays like that until I am resetting it. Maybe that is what Scratch makes sense, since I think in that way. I’m good at processing things to an outcome, I can see where things are going. [ERE4(Int)]
\end{quote}
Rina’s exemplification of how Scratch programming has helped her make sense of variable both in computing and in mathematics led to her developing a pedagogic approach she called *Thinking like a sprite* discussed in section 6.3.2 below. The connections that Rina made between the computational representation of variable and the algebraic representation were evidenced in the observed lesson detailed in the next section. [ERE2(Obsv)].

Rina taught Activity 5.1.3 (Figure 6.2) whose objective was to build a script for a generalised regular polygon and use variable to vary the side length and the number of sides. Rina discussed a solution with the children and stepped through the script to identify the value of the answer block at the end of each step.

For example:

- First, the script asks for a side length, 25 is typed in. The answer stores 25.
- Next, the script asks how many sides, 8 is typed in. The answer stores 8, the previously stored 25 is forgotten.
- The value of the answer block used in the repeat and in the move and turn expression is 8.
The discussion with the children led to the discovery that only one answer can be remembered in Scratch, and that the answer remembered will be the answer to the last question asked. Rina used this fact to introduce the concept of variables:

Today, we are going to teach it to remember some answers. We are going to call these variables. We are going to set some variables. This is quite exciting guys! We are going to go into a bit of Scratch that we have not even clicked on yet. In here you will see that there is one called data and you can make a variable. [ERE2(Obsv)].

Rina’s explanation of variable is brief and situated in its purpose e.g., a variable can be used to remember answers. The children did not question the introduction of variables and followed the instructions to create the variables with no observed difficulty. The task explained on the PowerPoint slide from the SM materials (Figure 6.3) is a procedural set of instructions and an opportunity to explore how the set block can be used to change the value of the side length variable.

![Figure 6.3 SM Activity 5.1.3: Instructions to create variables for side length and number of sides and explore their use](image)

However, in the next task which asked the children adapt the script to ask two questions and use the two variables to create a generalised script for a polygon; the children struggled. Rina had observed the children’s lack of progress during her class walk around. To intervene, she
stopped the class and focused their attention on a script she had created. The script is shown in Figure 6.4 and contained two key components:

- **asks** two questions and **set** variables to the **answer** for **number of sides**, and **side length**.
- a bug, as it draws part of a polygon. The script contained a **repeat**, a **move** and a **turn**, but the values were the default values.

Rina bridged to mathematics by asking the children to connect the Scratch language to the language of algebra when she said “If we were doing algebra and I would write no. of sides is a [She points to the no. of sides variable]. Which bit of this [Pointing to the repeat block script] would also be a?”

Figure 6.4 Rina's project containing two questions asked and two set variables and a simple polygon script to be edited.

Rina’s decision to use an unknown $a$ to represent the number of sides variable in Scratch demonstrates her knowledge of how she has connected variable in computing with variable in mathematics. Rina explained that she had been teaching algebra recently in mathematics, “That’s why I brought in “imagine this is $a$” as we’ve recently been discussing that $a$ could be different amounts, in the same way that answer could be”. However, focusing on the variable expressed in Scratch was not sufficient to support the children. The interplay between the variable quantity and the quantity that was represented in the geometrical relationship was
problematic for the children as they could not adapt the static values in the script to use the corresponding variable. The algebraic relationship between the geometrical representation in mathematics and in Scratch is discussed in more detail in the following section 6.2.3 Window on angle.

6.2.3 Window on angle

Rina’s lesson [ERE2(Obsv)] asked the children to create a regular polygon script to use two variables: side length and number of sides. During her walkaround Rina observed the children were struggling to make any progress other than creating the variables. Her first intervention directed the children’s attention to the geometric properties of the polygon represented in Scratch:

RE: To make a shape, I know I need to have a move and a turn, and I need to repeat that a few times to make a shape. Do you agree?
Children: Yes!
RE: We know that don’t we? So, this is my simple script to make a polygon. [She points to the script shown in Figure 6.5 below.]

![Simple script to draw a polygon using default blocks dragged from the Scratch palette](image)

Figure 6.5 Simple script to draw a polygon using default blocks dragged from the Scratch palette

RE: Do you agree?
Children: Yes!
RE: That polygon is probably not going to work. Because 10 repeats and only turn 15 degrees. How many degrees will it have turned all together?
Children: 150.
RE: So, is that going to be a full polygon?
Child: No!
RE: Right, so I have asked how many sides and I am telling it to set the number of sides to whatever that answer is. So, I’ve got this variable here, the number of sides.
**RE:** Which bit of the script is the numbers of sides, which bit of the script will always be the number of sides on my shape?

*She waits for many hands to be raised.*

**Child:** I think the number of sides will be 6?

**RE:** No, I have not decided how many.

A child had suggested a specific value for the number of sides rather than using one of the variables, a common issue when developing a concept of variable (Küchemann, 1978). The lack of children responses, (unusual for the very vocal and engaged class), suggested that the children were unclear on how a regular polygon script can be adapted to show the relationship between the variable and the structural property of the polygon. Rina then rephrased the question and asked the children to consider which part of the script would be the number of sides. A child responded, “the move”, an incorrect response since the move block represents the length of each side. Rather than responding to the child that they were incorrect, Rina chose to represent the child’s suggestion of the move as the number of sides:

<table>
<thead>
<tr>
<th>Dialogue</th>
<th>Activity on flipchart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RE:</strong> So, if I was to draw a shape and my move is 10. [She draws a line as she says this]. Is that the number of sides?</td>
<td>![Line on flipchart]</td>
</tr>
<tr>
<td>[The children remain quiet.]</td>
<td></td>
</tr>
<tr>
<td><strong>RE:</strong> Hmmm. My move is 3. [She draw a line smaller in length]. Is that the number of sides?</td>
<td>![Line on flipchart]</td>
</tr>
<tr>
<td>[The children remain quiet.]</td>
<td></td>
</tr>
<tr>
<td><strong>RE:</strong> So, it’s not the move is it?</td>
<td></td>
</tr>
</tbody>
</table>

Rina’s representation of **move 10** and **move 3** on the flip chart drawn as two straight lines were not sufficient for the child to realise their error. Rina continued her explanation:

<table>
<thead>
<tr>
<th>Dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RE:</strong> If I do this: [She draws a triangle].</td>
</tr>
<tr>
<td><strong>RE:</strong> This is what my sprite is doing when we draw a triangle…</td>
</tr>
<tr>
<td><strong>RE:</strong> Which bit of it: If I go: <strong>move</strong> and <strong>turn</strong>, <strong>move</strong> and <strong>turn</strong>, <strong>move</strong></td>
</tr>
</tbody>
</table>
and turn. [She puts emphasis on the pen moving and turning.] Which bit of it is the number of sides?

**Pupil:** I think it is the turn!

**RE:** The turn? So, if I want my shape to have 3 sides, [She traces the sides of the shape with her pen] I’m going to move and turn 3 degrees? Is that right? This is quite surprising!

At this point of the explanation the children were not following and were unable to overcome the conceptual obstacle. Rina continued her discussion with the children, however there was only one option left:

**RE:** Rihann? We only have one option left!

Rihann: It’s the repeat.

**RE:** It’s the repeat! If I do **move** and **turn** 3 times, [Draws out a second triangle emphasising the 3 move and turns.]

**RE:** My **number of sides** is my **repeat**. [She updates the script to show this step].

**RE:** Which bit of that is going to be my side length?

**RE:** Everybody look at this one, which bit of that is how long my side length.

**RE:** It could be 100 long or 5 long or 50 long. [Draws out 3 straight lines as she says each number].
In summary, the episode showed how Rina had to think on her feet (e.g. to think clearly in times of stress, especially when forming a solution to a pressing problem) responding to the problems to resolve the children’s difficulty with the relationship between the defined variables and the corresponding static quantity in the general polygon. She exemplified the beetle’s movement on the flipchart and made a connection between the length of the line (a geometric property) and the move block in the script which moved the beetle. The representations were not sufficient for the children to be able to explain the relationships between the variables in the scripts and their effect on the regular polygon script. However, with respect to Rina’s MPTK, the episode demonstrated that Rina was able to make connections between the geometric structure and variable concepts in computing, and the concepts in mathematics and was developing pedagogical approaches through her use of multiple representations to support the children with their developing computational and mathematical knowledge.

6.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

In this section, I present the two themes which emerged from analysis across the teacher data. Within each theme, I will illustrate how Rina characterises the theme and explain the relationship between the mathematical idea and the programmatic representation (see Table 4.5 on page 122 for classification of each theme).
6.3 How mathematical ideas mediate and are mediated by engagement with the
ScratchMaths curriculum

6.3.1 Thinking with your maths head on

Investigation two of the Fireworks microworld had an objective to build different sized
rectangles and rectangle patterns to explore mathematical similarity. Activity 5.2.2’s task as
shown in Figure 6.6 was to create a script which used two variables: height and base to draw
a rectangle. In the observed lesson [ERE6(Obsv)] the children created the script as instructed.
However, a surprising result had occurred for many children when they ran the script; a line
was drawn not a rectangle which they expected.

![Figure 6.6 SM Activity 5.2.2: Altering rectangles](image)

Rina observed the children struggling to interpret this output and stopped the class to enable a
whole-class discussion. Rina created and displayed a general rectangle script as shown in
Figure 6.7 below. She explained the steps that she had taken to create the script, e.g., created
new variables called base and height and put each into the definition for the number of steps
she wanted the beetle to move. She then ran the script and asked the children to explain what
they had noticed:
RE: Oh. What’s happened?

EN: It’s the base.

RE: Why is En saying it’s the base? Why am I not getting a rectangle anymore?

RE: Because when I take out the base [she removes the base variable from the script which leaves 10 behind and runs the script]. I have a rectangle, it is a bit of a squishy rectangle, but it is definitely a rectangle. But if I put in base [she inserts the base variable back in]. I don’t get a rectangle.

RE: What’s happened? Annas?

Annas: I think it’s because, see where the left corner, it says the base is 0. So, it moves zero.

RE: Ahh, so at the moment my base is how many steps?

Sarah: 0.

RE: 0, so if you draw a rectangle with no base, what is it?

Child: A line.

RE: A line, isn’t it. It is two lines going back on itself.

The classroom dialogue illustrates the theme of *Thinking with your maths head* e.g., that Rina encouraged the children to explain the output displayed on the screen from a mathematical perspective. She demonstrated that she has connected her programming and mathematical knowledge when she explained that the line drawn by the beetle is a rectangle with a base of 0, rather than dismissing the line as a Scratch bug. Scratch has a limited screen output for drawing; the stage size is 480 pixels wide x (-240 to 240) and 360 pixels tall (-180 to 180). Anything drawn outside of this range can produce an unpredictable output. Rina encountered this Scratch limitation in her previous lesson [ERE5(Obsv)] and demonstrated how her programming and mathematical knowledge were connected as explained in the following episode.
The pupils had drawn towers of rectangles with a fixed base, and an increasing height which grew each time the script was run. When the script was run repeatedly the rectangles height quickly became greater than the 360 pixels of the screen. The result is a broken rectangle, drawn without the two short sides as shown on the left hand side of Figure 6.8 below.

![Figure 6.8 Rina’s screen to show the result of a rectangle drawn of height 360 which goes out of screen bounds](image)

Rather than dismissing the screen output as an error, she used the pedagogical approach of explain and bridge to support the children with the development of their mathematical reasoning and to bridge between the computing and mathematics contexts. Rina guided the children to note that the height variable was set at 1320 when the script was run, a value that was too big to draw, and consequently explained that they needed to reset the variable to a smaller starting value.

Developing mathematical reasoning is also demonstrated in Rina’s use of an unplugged task in the observed lesson [ERE6(Obcv)]. The unplugged task shown in Figure 6.9 used a generalised rectangle script of two variables, height and base to make increasing rectangle patterns. The objective of the task was to match each script with its output script by envisaging and
explaining. Rina encouraged the children to use mini-white boards to capture their thinking as they reasoned through each script. She asked:

![Image of the activity](image.png)

**Figure 6.9 SM Activity 5.2.2: unplugged activity to match rectangle patterns with their corresponding scripts.**

**RE:** C and D look quite similar to me. Are they the same?

**Child:** No.

**RE:** They look the same to me, Drishti.

**Child:** Look at the base, one of them changes by 10, and one of them changes by 5.

**RE:** What changed by 10 and by 5?

**Drishti:** The width.

**RE:** So, they have the same height, same base, same number of repeats, both drawing rectangles, change height by 10, change height by 10, change base by 10, change base by?

**Child:** 5!

**RE:** 5! That’s a little tiny difference, isn’t it. But how does it affect the drawing? Drishti?

**Drishti:** There’s a smaller gap.

**RE:** Between?

**Drishti:** The bases. The first one has the same… as that one, the height is by 10, but the other one is smaller.

**RE:** If you look at this one [she points to the first one], each time it increases by 10. Is that increase of 10, the same as this increase of 10, [she points to the fourth one]?

**Child:** Erm.
6.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

RE: The same?
Child: Yes.

RE: Yes. Can this gap be [she points to the second one] an increase of 10?
Child: No.

RE: Is it more than 10 or less than 10?
Child: Less.

RE: So it must be the one that changes by? [she points to the fourth one]
Child: 5!

What is apparent from the analysis of this episode is Rina’s persistence to obtain reasoned explanations from the children without using the computer to run the scripts. She is deliberately rather obtuse in her approach when she asked, “they look the same to me!” and prompted the child to reason their response further. Her use of the 5E pedagogical framework through her use of explain, envisage and bridge provided opportunities for her to develop children’s mathematical reasoning skills. Rina explained that she had decided to adapt the short, unplugged activity into a paired discussion activity, an example of exchange, to provide an additional opportunity for children to talk to one another and to explain their thinking [ERE5(Int)]. Her use of the 5E pedagogical framework and reasoning step by step has benefitted, in her opinion, the children’s mathematical reasoning and exemplifies the Thinking with your maths head on theme of this section:

I think this has helped their arithmetic as they are not doing arithmetic for the sake of a process, they understand the steps more. I can say go back and find your error if you have made a mistake, they are a lot more resilient going back through it.

I’d say, you can click rectangle and it makes a rectangle or it doesn’t make a rectangle, or have you got the answer, or not got the answer right. In terms of written methods, it is understanding each step which is exactly what that is… I guess they can reason about arithmetic more, it has helped their arithmetic because they can reason about their written methods. [ERE6(Int)]

6.3.2 Thinking like a sprite

In this final subsection I present Rina’s use of the simile Thinking like a sprite in three different contexts. See Section 4.3 for exemplif which arose from comparative analysis across the data.
6.3.2.1 Thinking like a Digit sprite

In this episode [ERE1(Obsv)], Rina used with the Place value microworld in the context of time to Build a Stopwatch. Before the observed lesson, Rina had worked on building stopwatches with the children. Preparing for the lesson she noted multiple bugs when she reviewed children’s stopwatch projects. Her response was to create a broken stopwatch project that contained bugs the children had made and to debug the stopwatch in a whole-class activity. First, Rina ran her bugged Scratch project. The stopwatch displayed the following output 00:00, 00:01, ..., 00:59, 00:60, 00:61, ..., 00:99. The sequence of numbers displayed on the screen show that the ten seconds and one seconds sprites count up in ones from 00:00 to 00:99 and then both sprites change to 0. Watching the digits move in real-time, gives the observer (with the assumption that they have seen a stopwatch before, and have knowledge of how the digits change!) a sense that something is wrong. The children spotted that the one minute sprite was not changing at sixty seconds and suggested that the ten seconds script be updated to if costume # = 6, rather than if costume # = 10. Rina updated the script, as shown in Figure 6.10 below and re-started the stopwatch. She asked the children to explain what they observed.

Figure 6.10 Rina's screen displaying the broken stopwatch and the code inside the 10 seconds sprite.

**RE:** Are you ready? I’ve skipped it ahead to 50.

[She runs the script, the digits change to 00:50, 00:51, 00:52, ..., 00:59, 01:00, 01:01. She leaves the stopwatch running.]

**RE:** What do I need to change, what do I need to debug? Neham?

**Neham:** The 10 second sprite is not resetting.

**RE:** Why is not going back to 0 when it gets to 59, Noor?

**Noor:** Because you have a button…

**RC:** Have a think and I’ll come back. Think of your sentence first.
6.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

RE: Youseff?

Youseff: The thing that’s gone wrong is, on the board it says broadcast add one minute and wait. I think you used the wrong block. It should say broadcast add one minute.

RE: Alright so I’ll swap [she renames the broadcast message to be add one minute], broadcast add one minute, OK, right then. Reset, let’s go again, see what happens.

[She runs the script and the same error occurs. Changing the message sent was not required.]

RE: People are disagreeing. Drishti what do you think?

Drishti: I think you have more costumes, you have to hide the 6, 7, 8 and 9.

RE: So, what do I want my next costume to be. After 59, what do I want my next tens costume to be? Wasima?

Wasima: Erm.

RE: What comes after 5? Shout it…


RE: 6? Do we have 6? At the moment I’ve got 6 on there, but what should come after 5?

Wasima: Zero.

RE: Zero, so let’s get rid of these.

[She deletes costumes 7, 8 and 9 from the ten seconds sprite.]

RE: Right, do I need a 6?

Child: No.

RE: Let’s have a look. [She runs the script, the tens second sprite changes correctly]. Amazing!

The dialogue illustrates how Rina drew on the children’s knowledge of time to make connections with the costumes that are available to the sprite. She used multiple representations and bridged between the contexts of time represented by the broken Scratch place value model and the context of time in mathematics. The costumes of the sprite in the place value model were connected to the digits available in the mathematical representation of time. This type of mathematical reasoning and discussion of how time can be connected to a place value model is not something that is easily achieved without the use of Scratch. Rina explained that typically time is taught through facts, such as there are sixty seconds in minute, or half past four is the same as 4:30 on a digital clock or using a stopwatch to record time. The ScratchMaths activity of building and expressing a stopwatch in programming provided opportunities for Rina to engage with concept of time in the context of a place value model. By thinking like a digit
sprite, the children were supported to bridge effectively between the programming context and the context of mathematics to enable them to think deeply about place value structure.

6.3.2.2 Thinking like a Beetle for variable

In this episode from the observed lesson [ERE3(Obsv)] the children were working on building skyscrapers of towers of different sized squares as part of the polygon fireworks investigation (see section 4.2.2). The first part of the task required the children to draw a square with a variable for side length as shown below. The children had completed this task quickly by replacing the number in the move block with the variable side length.

![Defined square block using side length variable to control move](image)

However, they were struggling to create the script to draw a tower of ten squares. The children became stuck when moving the beetle to a position to start drawing the next square. Rina intervened when she noticed the children had an unexpected output when they ran their script. Figure 6.12 below shows exemplars of children’s towers of squares drawn with side lengths of 5, 10 and 15. The tower of side length 10 (the middle tower) looks to have displayed correctly. The bug is a result of not expressing the relationship between the side length and how far the beetle must move to start the next square i.e., use the side length variable in the move block. The following dialogue is taken from Rina’s intervention:
6.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

Figure 6.12 Scratch exemplar scripts to draw towers of squares with side lengths of 5, 10 and 15

**RE:** I’ve noticed a lot of people’s problems… When you draw a square, you need to find out where does your sprite stop. Mine just stopped there.

*She draws a square on the flip chart and indicates with a dot to represent the Beetles finishing location. See Figure 6.13 below.*

Figure 6.13 Rina demonstrating the starting and finishing place for the beetle on the flip chart

**RE:** If I wanted to start drawing my next square on top of it. *[she points to the top left corner of the square].*

**RE:** How far does it need to move? I’ve got it here *[pointing to the dot]* and I want to start drawing there *[pointing to the top left corner of the square].*
EN: The same.
RE: The same as what?
Child: The same as the side length.
RE: Yes, the same as side length.
RE: I’m going to use some algebra, because we’ve been using in maths. I’m going to call this m for Marion.

[She labels the left side of the square as m.]
RE: So, my side length is m [she gestures moving from the bottom left corner to the top left corner], that’s my variable. In yours it is not called Marion, it’s called side length.
RE: Sorry Marion you’re out!

[She crosses out the m and writes side length].
RE: So, I’ve moved the side length. How far do I move in this direction?

[She draws a block and writes sl into the block see Figure 6.14].
Children in unison: Side length.
6.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

![Image of Rina's flipchart showing a square, four labelled sides marked with length sl and a move block with sl inside it.]

Figure 6.14 Rina's flipchart showing a square, four labelled sides marked with length sl and a move block with sl inside it.

**RE:** Every time you go around. So, if I stop here, EN just said I need to move the same. So what do I need to move?

**Children in unison:** Side length!

**RE:** So, you need to use that side length variable in another place in your script, so make sure that every time it draws, it does it.

The episode shows how Rina connected multiple representations to make connections between the variable **side length** as expressed in the programming environment, a property of the square, and as a variable sl in algebraic notation. Rina moved fluidly between each representation; she expressed the beetle movement using the variable, she marked the length the beetle moved in the diagram, and she labelled the length of the square using algebraic notation. In the consequent Maths lesson (not observed), Rina explained that she strengthened the connections between the programming and mathematics. I had just finished collecting data for Sally, her partner Year 6 teacher and Rina excitedly called me into her classroom. She wanted to show me something that she had been working on with the children; they had been **Thinking like a sprite.** On the board was displayed some worded problems and beside each had written a mathematical expression of the total cost for each situation. Rina asked the children to **explain** what they had been doing.
The children explained that to write the expression using algebra they had to think like a sprite and consider what question the sprite would ask. For example, in the first situation, the beetle would ask “How many tickets?” and this helped them to know what the variable quantity was. In this example, the variable $t$. After identifying the variable in Scratch, they were then able to write the expression of $25t + 6$ using algebraic notation. Rina had organically developed thinking like a sprite as a pedagogical strategy to effectively bridge between the Scratch programming environment and writing algebraic expressions in mathematics. Using the beetle sprite as an object to think with was necessary for the mechanism of bridging to be effective. The object had specific functions available such as asking a question, and automatically storing the answer to the question as a variable. Thinking in this way supported the children to make sense of writing mathematical statements as expressions. Expressing mathematics in programming played an important role in enabling the children to move from the context of Scratch to the context of mathematics.

6.3.2.3 Thinking like a Beetle for angle

The final context of thinking like a sprite is situated in Rina’s lesson [ERE5(Obsv)]. Rina was teaching using the Fireworks microworld to program growing rectangle patterns as shown in Figure 6.16 below. The task provided a context to use a variable **height**, for a general rectangle with a fixed base. The two patterns require the variable height to be operated on using the change by block. In the first pattern the starting and finishing position is the bottom left-hand
corner. However, the second staircase pattern is more challenging as it required the child to consider where the beetle must move to before starting to draw the next rectangle.

Figure 6.16 SM Activity 5.2.2 – Altering Rectangles

During the lesson Rina circulated the room and checked the scripts the children had made, by discussing with each child and observing their output on their screen. She observed that many children found the second pattern problematic since they had not made any progress beyond the vertical tower. Her intervention was to demonstrate how the beetle moved by playing beetle in front of class as shown in Figure 6.17.
Case study 2 – Rina (RE)

Figure 6.17 Rina acting out playing beetle to draw the growing rectangle pattern using a variable for height.

The dialogue from the episode:

**RE:** I am beetle. How am I going to draw this shape? Tell me what to do.
**RE:** You all got as far as the first one, so you can start me off. Shaima?
**Child:** First you have to draw a rectangle and then…
**RE:** I can do that, I know a rectangle is move turn, move turn, move turn, move

[She steps the moves out playing beetle.]

**Child:** And then turn 90 degrees.
**RE:** Is that what we want? I was facing this way when I finished, do I need to turn 90 degrees? What do I need to do?
**RE:** I want you to have a think, get somebody stood up for your team and work it out.

The episode suggested that Rina used the thinking like a beetle pedagogical approach whilst thinking on her feet, to support the children with representing the geometric properties of the rectangle in Scratch. The pedagogic approach of playing beetle was encouraged as part of the 5E pedagogic framework which underpins the ScratchMaths curriculum although not explicitly stated within the teacher material for this activity. Rina’s use of playing beetle was effective in moving the children forward to complete their horizontal towers of rectangles, and to bridge between geometric ideas in mathematics and their expression in programming.
6.4 Teacher pedagogic practices

In this section, I identify and exemplify the kinds of whole-class activity types utilised by Rina as she taught using the Place value and Fireworks microworlds. I then analyse the quality of Rina’s use of orchestrations and the adherence to the ScratchMaths curriculum through use of the adapted TRU framework for teaching mathematics through programming as discussed in section 3.6.4.

6.4.1 Whole-class orchestrations

Throughout the six observed lessons, Rina used Scratch live in front of the class for at least 50% of the lesson’s teaching time. In each phase of the lesson, Rina utilised most of the whole-class activity types. Table 6.2 provides an exemplification of each of the types of whole-class orchestrations.
Table 6.2 Whole-class activity types, their characterisation and frequency observed across the six observed lessons

<table>
<thead>
<tr>
<th>Whole class orchestration type</th>
<th>Characterization of whole-class orchestration</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical-Demo</td>
<td>Rina used a Scratch technical demonstration once in the observed lessons. She demonstrated to the children a feature of Scratch which displays the value of variables to the screen. In all other lessons, Rina did not provide any technical demonstrations, which suggest that she had used and explained the blocks in previous (and unobserved) lessons and did not require further technical demonstration.</td>
<td>Rarely</td>
</tr>
<tr>
<td>Discuss-the-screen</td>
<td>The <em>Discuss-the-screen</em> whole screen orchestration was used most frequently by Rina and in each lesson. Her most frequent pedagogical approach was to ask the children a question, and then ask the children to <em>Explain</em> by saying “Why?”. Before Rina commented on the child’s response, she would ask the children if they agreed or disagreed with the response, and again would ask the child to explain why, rather than accepting an agreement. Rina explained that the school had a focus on oracy and have been drilled for two years to use sentence starters when explaining, for example using “the reason that I agree is because…” [ERE5(Int)]</td>
<td>Frequently</td>
</tr>
<tr>
<td>Explain-the-screen</td>
<td>This orchestration focused on making connections between programming and the mathematical content. Rina used this frequently to connect the coordinates used within the Scratch environment to the coordinate plane as children had used in mathematics, an example of the 5E bridging.</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Link-screen-board</td>
<td>This orchestration was utilised when Rina used a flip-chart to draw diagrams, representing on paper the output of the Scratch stage so that she could make additional annotations and labels. For example, making on a number of steps that the beetle had moved on the screen on a square drawn on the flipchart.</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Spot-and-show</td>
<td>Rina utilised this orchestration to shape the place value lesson [ERE1(Obsv)] observed. In advance of the lesson she had created a broken timer based on the bugs she had observed in the previous lesson. The children were asked to spot the mistakes, explain why the bug had occurred and then suggest a fix before working on the solution themselves. An example of Spot-and-show is given in the episode elaborated below.</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Sherpa-at-work</td>
<td>The Sherpa-at-work orchestration was used in two of the observed lessons. In both lessons, the ‘Sherpa’ was the teacher, Rina, being guided by the children, in response to her request of “Tell me what to do!”. This represents a different usage than in the wider literature. When a child is imprecise, for example “You need a turn block.”, rather than selecting any turn block, Rina asks a clarifying question “Which one, there are two of them?”. In this orchestration Rina encouraged the children to call out if they do not agree with what she was being told what to do by the child. At the end of the orchestration and before the completed script was executed, Rina confirmed with the children whether the script was correct by envisaging the output as they read through the blocks displayed on the screen.</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Board-instruction</td>
<td>The Board-instruction orchestration was used in all of Rina’s fireworks microworld lessons but not observed in the one place value microworld lesson. The orchestration was frequently used at the start of lessons to recap previous work, sometimes with a screen shot of static code displayed rather than live use of Scratch. Children had access to the computer to support their recall and explanation of their work. Rina used this orchestration when discussing and explaining the unplugged ScratchMaths activities which do not require the use of the computer.</td>
<td>Frequently</td>
</tr>
<tr>
<td>Link-multi-modal-representation</td>
<td>Rina used the Link-multi-modal-representation when she played Beetle in two different ways. In the first example, she explained how to draw a tower of variable sized squares. Rina played Beetle by using her pen to represent the beetle and drew on the flip chart as she moved the beetle pen according to the script that was displayed on the screen. The script required the use of a variable in two places within the script, in the side of the square, and in the move along the side of the square, in order to begin the next square of the tower. Using the flip chart, Rina was able to exaggerate the distance between the squares if the beetle did not move using the same variable. In a second example, she physically played Beetle, acting out the instructions from the children to support their thinking and create a tower of increasing height rectangles drawn horizontally.</td>
<td>Sometimes</td>
</tr>
</tbody>
</table>
A first observation of the table is the low frequency of the Technical-demo orchestration. In the observed lessons Rina and the children used Scratch fluently, she did not search for blocks or explain the meaning of the blocks as would be appropriate for the introduction of a new block. The blocks used in the lessons had been introduced previously. Rina had taught the ScratchMaths curriculum the previous year with the same group of children. The curriculum was designed to introduce computational thinking skills and the programming microworld in the first year, and to then use the developed skills to explore mathematical ideas in the second year. Rina often asked the children where to find a specific block, or to explain how a block operates, to support the children’s developing instrumental genesis of Scratch.

A second observation of the table is the Rina’s frequent use of the Discuss-the-screen and the Board-instruction whole-class orchestrations. Discuss-the-screen was enacted by Rina most frequently in every lesson. She utilised discussing the screen in two different ways. In the first way Rina held a whole-class discussion focussed on a short script displayed on the interactive white board. Rina asked questions of the children, whilst pointing to a block or several blocks to explain what each block would do. The children were also asked to predict or to envisage what the outcome on the Scratch stage would be when the script was run. In the second way, Rina used live coding to and narrated her actions as she dragged out a particular block or asked the children where a block was located. An example of Discuss-the-screen from an observed lesson is elaborated in section 6.4.3 below.

The Board-instruction whole-class orchestration was used by Rina at the start of lessons to encourage the children to recall their activity from the previous lesson. The children either had access to the computer to review their previous work or were asked to discuss in pairs what they had been doing before sharing with Rina. The high frequency can also be partly explained by Rina’s use of the unplugged ScratchMaths tasks and the ScratchMaths PowerPoint presentations. Unplugged ScratchMaths activities are designed to be used away from the computer, though the computer can be used to test out predictions. Children were given a print-out of the PowerPoint slide to discuss in pairs. Rina then asked the children questions to explain their thinking for each problem displayed on the PowerPoint slide on the board. Discussing the purpose of the PowerPoint slides, Rina explained that she doesn’t create additional notes for the lesson and that the PowerPoints are her notes for the lesson plan:
I just look at the [PowerPoint] slides and do it myself. Don’t feel I need the Teacher Notes because I learn better through doing it so I look at what you’re asking them to do and quickly do it myself. [Asked whether she uses the Teacher Notes at all] Yeah, I look at that at the start of a unit and think do I remember that and have I covered that and then ignore them! [ERE2(Int)]

6.4.2 Spot-and-Show orchestration

In this section I provide additional detail of the episode [ERE1(Obsv)] introduced in section 6.3.2.1 for Thinking like a Digit sprite. The episode used activity Build a stopwatch from the Place value microworld. Before the lesson started, Rina had reviewed the children’s Scratch projects and identified bugs in the children’s stopwatch project. She then built her own bugged stopwatch project and used the project to debug in a whole-class activity and fixed the bugs. However, a child’s project was still not working. Rina displayed the child’s work on the screen as an opportunity to debug as a whole-class. She ran the script, the stop-watch displayed 00:01, 00:02, ..., 00:09, 00:10, ..., 0:59, 00:00. Rina exclaimed to the children: “That’s not supposed to happen is it? Talk to your partner, what’s going on with this script?”. The children were given a minute to discuss the unexpected outcome. Rina then stopped the children’s discussion and asked them to explain what they had observed. On the screen was displayed the script for the one seconds sprite as shown in Figure 6.18.

![Figure 6.18 Script for one seconds sprite in the stopwatch activity](image-url)
RE: When I click the sprite, it waits one second, changes the costume until its costume number 9. What happens if its costume number 10?
Pupil: It adds 1 to the ten seconds.
RE: What does it actually do? It sends out a?
Pupil: Broadcast message.
RE: Whose listening for that broadcast message?
Pupil: The tens second is listening for that and it hears that message and it changes to next costume.
RE: Is there a problem with anything Zara says?
Pupils: No!
RE: Is there anything wrong with my script on the screen [The 1 seconds sprite] because I can’t change any of this to fix the problem. So, where might my problem be, Drishti?
Zara: In the ten seconds sprite.
RE: So as Zara did, have a look through this. [She selects the 10 seconds sprite]. What’s going on there? Have a read through this, something is not right! Let’s watch again what happens, to see if you can work out what is happening.

In the episode Rina encouraged the children to focus on the observed behaviour of the stopwatch and supported them to explain what the expected behaviour should be by drawing on their mathematical knowledge of time. As a class, they then verified that the script for the one seconds sprite was working correctly and confirmed that a broadcast mechanism was in place. However, they identified that the when I receive (the carrying/exchanging) mechanism was not programmed in the ten seconds sprite and found the bug in the child’s work. Rina continued to work in this way with the children throughout the lesson to support their debugging approaches. She encouraged the children to:

- observe and spot what happened,
- discuss in pairs to explain,
- share as a class how to resolve the bug,
- resolve the bug.

Rina explained that this approach had developed from the school’s focus on good mistakes, an initiative whereby it’s OK and is actively encouraged to make mistakes. She adapted and applied this approach to her ScratchMaths and her Maths lessons as discussed in her post-lesson interview.

We have a big school focus on good mistakes. This is the way I'm thinking not just in Scratch but also in Maths and debugging in English. I had a look through what they were doing last week… and what they were still struggling with. But that's the whole confidence and arrogance thing, I think that comes from, I know I understand it, and a lot of people don't, so I can see that mistake, and I think that not everyone necessarily would. [ERE1(Int)]
6.4.3 Discuss-the-screen orchestration

Rina used live coding for at least 50% of each lesson. She used Scratch in real time, live in front of the whole-class. Rina dragged out blocks from the Scratch palette, connected blocks together, updated values, and ran scripts. She used the interactive white board directly or sat at the nearby computer as shown in Figure 6.19 below. In this episode [ERE2(Obsv)] from Module 5 Investigation 1 - Polygon fireworks, Rina had developed a script with one bug which she displayed on the board. The script (Figure 6.20):

- asked a question,
- drew a polygon of that amount of sides,
- moved to x:10 and y:10 (the bug)
- repeated 25 times
Figure 6.19 Rina live coding using the computer sitting beside the IWB in the Discuss-the-screen orchestration. Figure 6.20 below shows the script she was building.

Figure 6.20 Rina’s script to draw a polygon for any number of sides containing a bug

Rina discussed the screen by pointing to her script and asking the children questions to explain each step of what was displayed:

**RE:** My script asks how many sides and waits. We use that answer, but what will the answer be? How are we using that answer? Have a look at my script?

**RE:** It asks how many sides and what does it do with that answer, Zara?

**Zara:** It asks you how many sides, and then it repeats 25 times, and then it repeats your answer.
RE: OK. Hmmm. What is this repeat doing, the 25? What will it do 25 times? Let’s look inside it.

[She points at the next repeat block under the repeat 25]

RE: Brad?
Brad: It will move 10 steps.
RE: Just that? It will move 250 steps?
Pupil: No.
RE: Mary?
Mary: It will make 25 shapes in the sky.

She then asked the children to explain the expression: “In here we also used answer, turn 360 divided by answer. What on earth is that all about?” A child responded correctly “if you divide 360 by your answer it will make the turn for the whole shape.” Rina confirmed the child’s explanation and ran the script. However, the output on the screen was that the beetle turned around and around but did not draw anything. The children suggested reasons for this, such as the pen was not down and also noticed that the go to x: 10 y:10 fixed the beetle in once place. To resolve the bug, Rina updated the code live to use a **pick random** block.

![Figure 6.21 Updated script to include the **pick random** for the go to x: y:](image)

The discussion of the screen involved:

- moving to the computer to edit the script (live coding) using suggestions from the children;
- running the script;
- moving to the board to point to features of the output and make connections with the script;
- asking discussion questions to explain the output.
Rina performed several cycles of *discuss-the-screen* to resolve the script’s bugs and produce a working script that drew 25 polygon fireworks in different places in the sky.

### 6.4.4 The TRU framework

In this section I report on the analysis of Rina’s observed lessons through use of the adapted TRU framework. The framework’s numerical scores represent a Basic (1), Proficient (2) and Distinguished (3) performance in each dimension (see Appendix A for the full scoring rubric). The results for each observed lesson are shown in Table 6.3 and represented as a chart in Figure 6.22 below.

<table>
<thead>
<tr>
<th>Date</th>
<th>Microworld</th>
<th>Mathematics through programming</th>
<th>Cognitive demand</th>
<th>Equitable access to content</th>
<th>Agency, Ownership and Identity</th>
<th>Formative assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/11/2016</td>
<td>Place value</td>
<td>2.5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>20/03/2017</td>
<td>Fireworks</td>
<td>2.6</td>
<td>2.6</td>
<td>3</td>
<td>2.4</td>
<td>2.3</td>
</tr>
<tr>
<td>05/04/2017</td>
<td></td>
<td>2.6</td>
<td>2.7</td>
<td>3</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td>05/06/2017</td>
<td></td>
<td>2.7</td>
<td>2.5</td>
<td>3</td>
<td>2.7</td>
<td>2.7</td>
</tr>
<tr>
<td>21/06/2017</td>
<td></td>
<td>2.4</td>
<td>2.4</td>
<td>3</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td>28/06/2017</td>
<td></td>
<td>2.5</td>
<td>2.5</td>
<td>3</td>
<td>2.6</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Table 6.3 Weighted averages for each dimension of Rina’s observed lessons using the adapted TRU framework for Maths through programming

<table>
<thead>
<tr>
<th>Date</th>
<th>Place value</th>
<th>Fireworks</th>
<th>05/04/17</th>
<th>05/06/17</th>
<th>21/06/17</th>
<th>28/06/17</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/11/16</td>
<td>3</td>
<td>2.5</td>
<td>2</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
</tr>
<tr>
<td>20/03/17</td>
<td>2.5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>05/04/17</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>05/06/17</td>
<td>2.5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>21/06/17</td>
<td>2.5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>28/06/17</td>
<td>3</td>
<td>2.5</td>
<td>2</td>
<td>3</td>
<td>2.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Figure 6.22 Chart to show weighted averages for each dimension of Rina's observed lessons using the adapted TRU framework for Maths through programming

The consistently high scoring of Dimension 3, Equitable access to content reflects Rina’s strong classroom management skills through well-established participation structures. Rina’s lessons consistently provided multiple opportunities for the children to discuss their mathematics or programming with one another, managed by Rina in tightly timed thirty second periods [ERE1-6(Obsv)]. Some children’s explanations were then shared in a whole-class setting. Rina was careful in the observed lessons to ensure participation from all children by circulating the class during their computer activity. There was no evidence during the observations of children disengaged with either the whole-class discussions or the individual computer work.

**Dimension 1, Mathematics through programming**

Rose engaged with the 5 E pedagogical framework across the observed lessons. She provided frequent opportunities in each lesson for the children to *Explore* scripts and to *Explain* their thinking. For example, in [ERE4(Obsv)] the children explored how they could set and change variables using the corresponding blocks, a concept which most of the class were unfamiliar with. They were then asked to *explain* what was the same and what was different. The resulting discussion allowed the children to *explain* their thinking and to share their developing understanding of variable.
Rina used *Envisage* as a strategy to encourage children to reason through a script before they ran it, or in her use of unplugged activities such as activity 5.2.1 shown in the Figure 6.23 below. Rina decided to adapt the task which had been designed to be a short class discussion, into a more substantial 30 minute paired activity. The children were provided with mini-white boards to capture their thinking. In the whole-class discussion [ERE4(Obsv)] Rina *bridged* to mathematics. She drew diagrams on the whiteboard as shown in Figure 6.24 to represent the beetle movement, labelled the squares as the variable side length changed, and expressed the final side length as an expression as show in the figure. Her decision to represent the activity, and how she *bridged* to mathematics were Rina’s choices, they were not provided as prompts within the PowerPoint slide. Rina explained that within the activity, she could see embedded mathematics and opportunities to develop children’s calculations for the final side length [ERE5(Int)].
Figure 6.23 SM Activity 5.2.1: Sequence of Squares

Figure 6.24 Rina’s white board to show the output of the script as the beetle moves to draw square patterns

The 5 E’s were not consistently present within each lesson; there were a few opportunities to bridge to mathematics as seen in the place value stop watch lesson [ERE1(Obsv)], and in the lesson which introduced variables [ERE2(Obsv)]. However, holistically from the scoring
rubric, **Distinguished:** The teacher uses orchestrations and makes meaningful connections between representations in programming and representations in mathematics (bridge).

**Dimension 2, Cognitive demand**

Rina used the Scratch Maths materials in every lesson. She explained that her planning process was to read the PowerPoint slides before the lesson, to try out the Scratch projects herself, or to **envisage** them in her head. She did not make use of the *ScratchMaths teacher guide*, which contained additional links to mathematics and suggested questions to further mathematical thinking. However, Rina adapted the ScratchMaths materials within several lessons. For example, the creation of the broken stopwatch containing bugs, the extended use of unplugged activities which increased the complexity of the provided tasks. Thus, from the scoring rubric, **Distinguished:** The teacher hints or scaffolds support students in productive struggle in building understandings and engaging in mathematical practices.

**Dimension 4, Agency, Ownership and Identity**

The amount of airtime in each lesson is typically shared between Rina and the children. Rina did not always call on the children who raise their hands first. The control of the dialogue was not always maintained by Rina in the classroom, for example Rina often sets up opportunities for the children to disagree or agree with something another child has said. Another example is when the children were provided an opportunity to share their programming work when stuck and then used their work to debug as a class. The following exchange illustrates an opportunity Rina used to resolve a child’s problem as a class:

RE: Let’s see if we can solve Shaima and Seema’s problem.

RE: EVERYBODY turn your chair to see the board!

*[Rina opens up their work from the Scratch interface, several pupils move to the carpet.]*

RE: What should we check works first? Malaika?

Child: The square block.

R: Let’s check. Does the square work?

*[She presses green flag, clicks the square block, and it draws a square]*.

RE: Does the square work?

Pupils in unison: Yes!
R: Then we have when this sprite is clicked, it’s going to do something 10 times, it’s going to ask what’s the side length, set the side length to the answer, draw a square, move 20 steps and turn 90 degrees.

B: That’s a lot.

R: I was just thinking that. Shaima and Seema what don’t we need to do 20 times.

Shaima: Ask the question!

R: So where could we put the question? I’ve taken them out of the repeat. Where could the question go?

Seema: Into the define square.

R: If we put it in define square, it would still be in this repeat, it would still be repeated 10 times. Mustafa?

Mustafa: You can put it on top of the repeat.

R: Can you see that if we put it out of the repeat so it won’t happen ten times?

From the scoring rubric, Distinguished: Students explain their ideas and reasoning. The teacher may ascribe ownership for student’s ideas in exposition, AND/OR students respond to and build on each other’s ideas.

**Dimension 5, Formative Assessment**

Children’s thinking is generally solicited and refined within the whole-class setting. Rina often asks the children “Why?” rather than just providing corrective feedback. She used children’s thinking typically to facilitate the completion of the specific task although she sometimes used a technique, called ‘Good mistakes’ [ERE1(Int)]. Good mistakes provided an opportunity for the children to make mistakes and then to share that mistake and to debug the script in the whole-class setting. The resultant discussion provided opportunities to develop children’s debugging and mathematical reasoning skills. The form of assessment from the scoring rubric lies somewhere between Proficient: The teacher refers to student thinking, perhaps even to common mistakes, but specific students’ ideas are not built on (when potentially valuable) or used to address challenges (when problematic) and Distinguished: The teacher solicits student thinking and subsequent instruction responds to those ideas, by building on productive beginnings or addressing emerging misunderstandings.
6.5 Summary

In this chapter, I have presented an analysis of Rina’s case data organised as i) windows on mathematical knowledge, ii) mathematical ideas mediated and mediated by the ScratchMaths curriculum, and iii) teacher pedagogic practices, to examine her mathematical pedagogical technology knowledge (MPTK). The analysis of Rina’s MPTK shows that she was able to spot the errors in the children’s pencil and paper calculations and able to explain that a child had a place value misconception in their understanding of the value of a digit in a two digit number. There was evidence to suggest that Rina was beginning to think about the place value models more generally than just within base ten, as she described the structure of the different place value models as a system of conditional checks, broadcasts and receiving messages, and acting on those messages. She demonstrated that thinking about time as an example of a place value system was a new connection because of teaching the programming representations of the place value models. Rina’s remarks on why column addition cannot be used to add times together unless you consider the rules differently than in base ten strengthen this claim. Rina used multiple representations to make links between variable as used in computing to store a value, and in algebraic notation to represent a known or unknown quantity in mathematics.

Rina developed children’s mathematical reasoning skills through her pedagogic approach of consistently asking children to explain their thinking as they processed step by step and each block in a script. She adapted unplugged activities to provide greater opportunities for children to reason and provide explanations. The data illustrated how she used a pedagogical strategy she called thinking like a sprite. The strategy which then shown in different contexts such as playing beetle at the board or in role play, using the costumes available to the sprite when explaining the time place value model, and how thinking like a sprite can help with identifying the unknown quantity when writing algebraic expressions for word problems.

Analysis of the lesson observations using an adapted TRU framework for mathematics through programming suggested a consistent and proficient use of the ScratchMaths investigations ((Dimension 2, Cognitive demand), and adapted materials to provide children with further opportunities to build understanding and engage in mathematical practices. Rina used the 5 E
Pedagogical framework consistently across the observed lessons, although not every lesson featured opportunities to use all of the E’s, rather than Rina not using a specific E. When she did use all of the 5E’s she used multiple representations as an effective way to bridge between computing and mathematics and made meaningful connections between representations in programming and representations in mathematics (Dimension 1 - *Mathematics through programming*). Rina created a classroom with high levels of meaningful mathematical participation from all students (Dimension 3 – *Equitable access to content*) through well-established activity structures and routines. Children in Rina’s classroom are provided opportunities to explain their reasoning, through shared airtime with Rina, and can offer agreement or disagreement with other children’s suggestions (Dimension 4 – *Agency, Ownership and Identity*). Rina had developed a strategy that makes coding mistakes to be “good mistakes”. She used the mistakes as opportunities for the children to debug each other’s work, or to debug special prepared projects which contain bugs to be found and resolved. (Dimension 5 – *Formative Assessment*).

Table 6.4 below provides a summary of the findings and their location within each section of the chapter where they are exemplified.
Table 6.4 Table to show summary of findings and location within the chapter for Rina's case study

<table>
<thead>
<tr>
<th>Window on mathematical knowledge for teaching</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window on place value</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Identified misconceptions in children’s place value tasks.</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Explained programmatic structure of Scratch place value model including broadcast &amp; receive.</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Developed connections between Scratch place value model and Scratch stopwatch model.</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Beginning to think about time as an example of a different place value.</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Developed connections between place value models in Scratch and own mathematical knowledge.</td>
<td>6.2.1</td>
</tr>
<tr>
<td>Window on variable</td>
<td>6.2.2</td>
</tr>
<tr>
<td>Explained costume # in the context of variable.</td>
<td>6.2.2</td>
</tr>
<tr>
<td>Used Scratch programming to help make sense of variable concept.</td>
<td>6.2.2</td>
</tr>
<tr>
<td>Developed connections between computational representation and algebraic representation of variable.</td>
<td>6.2.2</td>
</tr>
<tr>
<td>Window on angle</td>
<td>6.2.3</td>
</tr>
<tr>
<td>Moved between representations of geometric ideas: in programming, on screen and on paper.</td>
<td>6.2.3</td>
</tr>
<tr>
<td>Developed connections between geometric structure in mathematics and in programming.</td>
<td>6.2.3</td>
</tr>
<tr>
<td>How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum</td>
<td>6.3</td>
</tr>
<tr>
<td>Thinking with your maths head on</td>
<td>6.3.1</td>
</tr>
<tr>
<td>Interpreted scripts step by step to encourage mathematical reasoning.</td>
<td>6.3.1</td>
</tr>
<tr>
<td>Used unplugged activities to foster reasoned explanations and to envisage the output of programming.</td>
<td>6.3.1</td>
</tr>
<tr>
<td>Thinking like a sprite</td>
<td>6.3.2.1</td>
</tr>
<tr>
<td>Used the sprite and its costumes as objects to think with.</td>
<td>6.3.2.1</td>
</tr>
<tr>
<td>Used the beetle as an object to think with.</td>
<td>6.3.2.1</td>
</tr>
<tr>
<td>Developed Thinking like a sprite to make connections between algebraic and scratch representations for expression.</td>
<td>6.3.2.1</td>
</tr>
<tr>
<td>Teacher pedagogic practices</td>
<td>6.4</td>
</tr>
<tr>
<td>Whole class orchestrations</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Taught using Scratch live for at least 50% of the lesson.</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Developed bugged Scratch projects to debug live an example of the Spot-and-Show orchestration.</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Frequently used Discuss-the-screen orchestration</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Frequently used all of the 5 E’s to make connections between Scratch and mathematics. Bridge was most effective when using and linking multiple representations. [D1]</td>
<td>6.4.1</td>
</tr>
<tr>
<td>High level of engagement with SM curriculum [D2].</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Consistent classroom routines for children and teacher to engage with each other [D3].</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Used a “good mistakes” strategy to foster debugging skills. Children always asked “Why?” to explain their thinking. Children’s ideas used and developed [D4, D5].</td>
<td>6.4.1</td>
</tr>
<tr>
<td>The TRU framework</td>
<td>6.4.2</td>
</tr>
<tr>
<td></td>
<td>6.4.2</td>
</tr>
<tr>
<td></td>
<td>6.4.3</td>
</tr>
<tr>
<td></td>
<td>6.4.4</td>
</tr>
</tbody>
</table>
Chapter 7  Case study 3 – Sally (SE)

7.1  Introduction

In this chapter, I present a description and analysis of selected episodes from Sally’s lesson observations teaching with the Fireworks microworld, and her responses to the task-based interviews; both of which open a window onto her mathematical pedagogical technology knowledge. The case is structured into three sections i) Windows on mathematical knowledge for teaching, ii) Technology shaping and shaped by the mathematics, and iii) Teacher pedagogic practices. The relationship of the three sections to the research questions and mapped on to the conceptual framework is shown in section 4.3 of Chapter 4.

7.2  Window on mathematical knowledge for teaching

7.2.1  Window on variable

Sally first recalled encountering variable during secondary school and described an example where she used a variable voltage to create physical circuits. She explained that the context was very different to how she had used variables in computer programming such as for keeping the score in a game. She defined variable as “Data which can only hold one piece of information at a time, and this information can change depending on certain conditions” [ESE4(Int)], a very computer centric definition.

In the first observed lesson Sally introduced the children to the concept of variable in the lesson [ESE1(Obsv)] using the Fireworks microworld. She showed the completed Scratch project as shown in Figure 7.1 from the provided materials. The project when run asked three questions and set three variables for the corresponding values.
• How many polygons?
• What side length?
• How many sides?

The beetle then moves to different positions in the sky, changes the pen shade, and draws many regular polygons using the three variables.

Figure 7.1 Completed Polygon Fireworks scratch project

Sally introduced the project displayed on the IWB and explained that it was a completed example of what she wanted the children to do, i.e., build a script to draw polygon fireworks in the sky. She pointed to the main script which asked the questions and introduced the children to the variable blocks.

SE: These [she points at the screen], we’ve got an ask and a set, an ask and a set, an ask and a set. These are variables, ok? [She sounds unsure.]

SE: So, what I want you to do now, is look at how we might choose to vary some of those and actually decide, can you vary them?

[The class are silent]

SE: So, a variable is something that changes.

[She gestures her hands up and down]

SE: So, it’s a bit like our algebra that we are doing at the moment. We are actually using these in maths today. We are going to be looking at the variables in our maths lesson today. OK?

SE: I want you to think about, for this script, can we change any of our ask and sets? Talk to the person next to you, off you go.
The dialogue exemplified that Sally is telling the children facts about variable rather than explaining the concept more fully and is potentially an indicator of her weak knowledge of variable. However, she did create a tentative bridge between variable in computing and in mathematics. For the discussion task, she asked the children to discuss the script by asking “Can we change any of our ask and sets?” It is unclear what Sally meant by this phrase as is evidenced by the difficult to follow classroom conversation:

SE: Do we think that there is anything that we can change? Cos, we can ask different things… Do we think that we can change [she gestures her hands up and down] those questions?

[Silence for 5 seconds]

SE: Yes or No?

[Silence for 4 seconds]

SE: Daniel?

Daniel: I don’t think you can vary… because you need to work systematically. If you don’t do that, it’s going to ask you and it’s going to set it. For example, you can’t set the number of polygons if you don’t know the answer because you need to set it to the answer.

SE: OK, so the variables can we change or not? What do we think? Could we change number of polygons to answer?

[She gestures to the set block]

SE: Could we change that? What could we change it to?

Daniel: I’m not sure.

SE: OK.

SE: For this particular activity, the whole point of asking the question is to what? What does the sprite need to know? Aeris?

Aeris: The sprite needs to know the answer.

SE: Yeh, so therefore the variable will be what? The variable is what? Aeris?

Aeris: The answer.

SE: The answer.

Again, in this classroom dialogue Sally told the children, correctly, what she knows about the blocks “The sprite needs to know the answer, the variable will be the answer”. However, the facts were not drawn together to make a coherent argument about the content. Sally seemed to notice that the children were not following her explanation, so she transitioned the class to paired computer work. In the post-lesson interview [ESE1(Int)] Sally explained that her thinking at this point of the lesson was to try to ascertain what the children knew about variables and more specifically to identify what aspect they thought was changing within the script:
I would expect them to say the number of polygons would be a variable and therefore because it’s being set and it’s awaiting the answer, that then impacts the next question. And then looking in there, there are 3 variables. That would be what I would be expecting. But like I said, *I then stopped, I don’t know it, they don’t know it*. Let’s stop there. [ESE1(Int)]

Her response seems to suggest that she realised within the lesson that her knowledge of variable was insufficient to continue with the explanation. Sally confirmed this claim in a later interview when she was asked to reflect on her development when watched the video of the lesson episode. She explained:

> Looking back, my default position *[at that time]* was I kinda know, the kids sorta know, so if that’s happening, then something has got to be going right! NOW, looking back at that, I’m staring at that script, thinking “there’s so many blocks there” and I can see myself going “remember what point you’re trying to get out of this.

> I’ve done a thing I call “when someone is dragging you through the grass”. If someone was dragging you through the grass, and you don’t want to be dragged through the grass, you would be trying to grab bits of grass to stop yourself from being dragged Because, I’m trying to remember all the content, the subject knowledge, pedagogy that I don’t have. So, the subject knowledge at that point, I didn’t really have, about variables. I didn’t have any pedagogy at all, and I knew something about Scratch but not a lot. So, therefore to me, I was trying to grab any blades of grass to slow me down by saying things I knew about Scratch. [ESE5(Int)]

Sally’s choice of the blade of grass analogy, suggested that she is trying to use the knowledge that she had available to her, e.g., some knowledge about Scratch variable., to slow her down from being dragged through an activity that she did not have enough knowledge to explain. Sally had not attended any PD due to her appointment at the school being after the PD had been delivered. During the interview it also emerged that she had not seen the *ScratchMaths Teacher guide*. She had thought that the PowerPoint and the SM Scratch projects were what she was expected to teach from. She was excited to discover that the guide explained the subject content along with pedagogical approaches to support teaching of the SM activities. Over the next few weeks Sally worked though the *ScratchMaths teachers guide* for working with the *Fireworks microworld*. Due to the school preparing for the KS2 mathematics tests, SM lesson were cancelled, and I was unable to collect any data.

In the first SM lesson [ESE2(Obsv)] following the break Sally revisited the polygon firework script as shown in Figure 7.2. Notably, the script Sally chose to display is less complex than the one that she had used in the previous lesson. This project used one *ask* block to ask a question “How many sides” and stored the input in the *answer* block; an instance of a variable to control the number of sides of the polygon.
The children were asked to recall their previous lesson and used the project displayed on the IWB as a stimulus. Sally engaged in a whole-class discussion with the children who explained that they had used a beetle sprite to make polygons with different shapes and colours. Sally used this as an opportunity to redefine variable which she had attempted to explain in the previous lesson.

Last time, we weren’t sure what a variable was, a variable would be something that can change. [She gestures her hands up and down]. We talked about this in maths, didn’t we? A little bit before we did our tests, as something that can change, but the difference with this is that it stores information. So, it stores and holds that information. [ESE2(Obsv)]

Her explanation was accurate and indicated a stronger definition of variable in the computing context as well as demonstrating a connection to the mathematical context. Sally explained in the interview [ESE6(Int)] that the link to variable in mathematics is when you have an unknown and you are trying to find the value when you solve an equation. However, she had not considered variable in other areas of mathematics, such as a variable within a formula, or a variable within an expression which does not have a fixed value. Over the next few lessons Sally used and developed the concept of variable in Scratch in different computational contexts, such as setting a variable to a different value, operating on the variable inside a move block and changing an existing variable to a different value. Both contexts are illustrated further in sections 7.3 below. Notably in each of the observed lessons, Sally’s explanations to the children were more coherent and demonstrated how she had connected her knowledge of Scratch and of mathematics into more coherent argument.
Sally returned to the polygon fireworks algorithm in a task-based interview [see section 3.4.2.2] after she had finished teaching the fireworks microworld. The objective of the task was to build a regular polygon algorithm that would ask two questions for the side length and the number of sides and then use those values to draw a regular polygon. Sally was asked to ‘think aloud’ and explain as she worked on the task. Table 7.1 below describes the actions Sally took in Scratch and what she said as she worked on the task. The right-hand column suggests a description of the aspect of Sally’s MPTK demonstrated at each step.
<table>
<thead>
<tr>
<th>Thinking out loud</th>
<th>Scratch scripts</th>
<th>Descriptive analysis</th>
</tr>
</thead>
</table>
| **SE:** I would probably do the basics first, let me move these blocks out of the way.  
[She drags out a move a turn and a repeat and defines it polygon.] | ![Scratch script](image1.png) | The step suggests Sally has formed a connection between the structure of a square and the general structure of a regular polygon, e.g., a **repeat**, a **move** and a **turn**. |
| **SE:** I know that I need to change my degrees *the turn*, using the division block as that affects how many times I repeat because of the amount of sides.  
**SE:** So, I would put 360 divided by the repeat.  
[She points to the number inside the repeat]. | ![Scratch script](image2.png) | The step indicates that Sally can calculate the angle of a turn in a regular polygon as 360 divided by the number of sides. She has made a link between variable as the number of sides and the number inside the repeat block. |
| **SE:** So, it makes more sense to put in an ask and an answer…  
[She drags out an ask block, two answer blocks, and inserts them into the script.] | ![Scratch script](image3.png) | Sally has represented the relationship stated in the previous step using programming language.  
She has recognised that the quantity which represents the number of sides can be replaced by the variable answer, in both the repeat and in the turn. |
Sally runs the script twice, using the value of 5 and 8, she is happy with the output.

Sally has checked her script by running it, using the feedback of the beetle’s drawing to confirm that her script is correct. This step demonstrates a check and learn from feedback approach.

Table 7.1 Extracts of Sally thinking out loud as she worked on the interview based Scratch task

Sally’s progress in this part of the task is significantly more advanced than her first observed lesson which used a similar polygon procedure. In the task she had been able to make connections between the structure of a square as represented in Scratch and how that structure can be used to create a general procedure for a regular polygon. She had used the answer block to represent the variable quantity and to represent the quantity within the general polygon algorithm. The connections in her knowledge were not present in the first lessons when she was not able to build the procedure live, or able to explain the connections between variable and the geometric structure of the shape. In the second part of the task, she was required to introduce a second question of side length. Sally was less confident but nonetheless was able to demonstrate her development in her understanding of variable. Table 7.1 continues below.
<table>
<thead>
<tr>
<th>Thinking out loud</th>
<th>Scratch stage</th>
<th>Descriptive analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SE:</strong> So now we need the length of the sides.</td>
<td>![Scratch stage image]</td>
<td>Sally suggested that this probably would not work before she ran it but found it difficult to articulate the reason when prompted why. She had tried to <em>envisage</em> the output and said that the order was not right. She then ran the script to see what the beetle would do, again demonstrating her run, check and learn from feedback approach.</td>
</tr>
<tr>
<td>[She adds in two ask questions blocks, creates a side length variable and sets it to answer, and adds an additional move block using the side length variable.]</td>
<td>![Scratch stage image]</td>
<td>Sally needed support to identify the bug. It was suggested that she should step through the script block by block and to say her thinking out loud.</td>
</tr>
<tr>
<td>[She runs the script and notices that the output is not quite right. One side appears slightly longer.]</td>
<td>![Scratch stage image]</td>
<td>She stepped through the script and was able to <em>explain</em> and identify a bug.</td>
</tr>
<tr>
<td><strong>PS:</strong> Why did you create a new variable?</td>
<td>![Scratch stage image]</td>
<td>• The answer to the first question was forgotten. Sally noted that she would need a variable to store it.</td>
</tr>
<tr>
<td>[She creates a new variable called number of sides and drags it out. She drags out a set number of sides and drags the answer into the set box. She reorders the ask and set questions.]</td>
<td>![Scratch stage image]</td>
<td>She was prompted to update her polygon definition. Sally was then able to <em>explain</em> that the move side length steps (the cause of the longer side) needed to move inside the definition, a critical part of the reason why a variable had been created.</td>
</tr>
<tr>
<td><strong>PS:</strong> Why did you create a new variable?</td>
<td>![Scratch stage image]</td>
<td>Sally’s explanation indicated that she understood a purpose of setting user defined variables in Scratch is due to the limitation of the <em>answer</em> block which gets overwritten when a second question is asked. She explained that the variables would then be constant, i.e., not be able to be affected by another part of the script.</td>
</tr>
</tbody>
</table>
Case study 3 – Sally (SE)

SE: It will set it and hold it, [so] it then won’t be affected by ask and answer… I wanted to make sure that they stay the same through the script. I wanted them to be the constant.

[She updates the polygon definition to use the two variables.]

PS: Can you get the beetle to show the perimeter of the shape?

[She drags a say block from the palette, a multiplication operation, and replaces the holes with variables.]

SE: Perimeter is usually all the sides added together, but I know the number of sides times the side length would find the perimeter. So I’ve got the two elements [variables].

Sally found this part of the task trivial.

Her actions demonstrated she was able to operate on two variables and use the variable to express her verbal explanation of perimeter as a Scratch expression. This suggests Sally has created connections between the two representations.
In summary, Sally was able to make progress and complete the task with some support prompts to explain her thinking. Her success in the activity, the strength of her explanation at each step, and in particular her use of variables to express the perimeter, is an indicator of a change in her knowledge of variables in computing and their relationship to mathematics. It also highlights Sally’s knowledge about the way variable in computing works, specifically that the value of answer in Scratch is deleted when a new question is asked and how to get round this by creating and setting a variable.

### 7.2.2 Window on angle

An early observation of Sally’s teaching revealed that she was not aware of the accompanying ScratchMaths teacher guide. However, despite not knowing what she did not know, she planned and taught a ScratchMaths lesson using the PowerPoint slides. The PowerPoint slides were designed to be used by the teacher to provide structure and support to a lesson or as an aide memoire as discussed in section [4.2.3]. In the lesson [ESE1(Obsv)], Sally used the Polygon Fireworks investigation from Module 5. The objective of the investigation is to fill the London sky with regular polygon fireworks of varied sizes, sides and colours. At the start of the lesson, she displayed a slide of her own creation (Figure 7.3).

![Figure 7.3 Sally’s creates slide which asks the children to consider whith polygon is most challenging to draw in Scratch](image)

Given my own knowledge of Scratch and of prior research (see for example Clements and Sarama (1997)) this is an effective question to uncover children’s misconceptions of how the
beetle turns. A well-researched misconception when learning Turtle Geometry is to reason that the beetle turns through 60 degrees when drawing an equilateral triangle, rather than 120 degrees and could therefore be considered a challenge. The error arises when it has not been appreciated that the beetle turns through the external angle. An analysis of the dialogue between Sally and the class uncovered that Sally also held this misconception and illustrates how she dealt with it.

In this first part of the episode, Sally asked the children to explain the complexity of each regular polygon:

SE: Hamim, which one do you think would be the least challenging?
Hamim: I think the square.
SE: Why?
Hamim: Because you know all the sides are equal.
SE: These other regular polygons, are the sides not equal?
Hamim: They are, but errr.
SE: Do you want me to phone a friend, Aeris?
Aeris: The square is more basic to me, it doesn’t challenge your brain that much.
SE: Why does it not challenge your brain as much?
Aeris: Because there’s are only 4 sides that are equal.
SE: OK, Habiba?
Habiba: I think the square is the least challenging because you only need two instructions which is a move how many steps and then turn 90 degrees. And because there’s 4 sides, you’d need a repeat button which says repeat 4 sides.
SE: Mmm OK.

Sally attempted to refute the child who explained that the square is least challenging with her use of the mathematical fact that all regular polygons have equal sides. However, when another child offers a correct explanation for the algorithm for a square, she appeared unsure as to how to respond, offering a tentative ‘OK’. All regular polygons use only two instructions inside a repeat block, which would be a reasonable explanation to the children to explain that all the polygons are the same level of challenge. However, as the dialogue continued, it emerged that Sally was unclear herself on how the beetle turned and how the angle of turn could be calculated for a regular polygon.

SE: Who thought the triangle would be the least challenging?

[A few hands go up.]
SE: Oh, I thought everybody would say the triangle as it’s got the least amount of sides.
SE: What do we know about equilateral triangles?
Child: The sides are the same.
SE: And the?
Child: Angles.
SE: Angles. But what is different about the triangle compared to the others, think about the angle total?
[Children gasp.]
Children: It all adds up to 180.
SE: Excellent, and the others add up to?
Children in (unison): 360!

Unpicking the dialogue, Sally initially explained that the complexity of making each shape is related to the number of sides of the shape. This could be a valid explanation if the shapes had been drawn with direct control over the beetle, e.g., a move and a turn, a move and a turn, a move and a turn. I.e.e, the greater the number of sides, the greater the number of blocks. However, Sally chose to use the angle fact for the sum of the angles inside a triangle is as degrees as the difference in challenge. She is likely confusing interior and exterior angles sums as is shown by her board diagram shown in Figure 7.4 below. She does not make clear which angle she is referring to within the shape but marks a corner. Both angle facts she stated are true separately, but when used to explain the angle of turn for a regular polygon algorithm, they conflict with one another. Notably, Sally did not apply the formulae for a triangle when she calculated the angle of turn for a pentagon and a square, which would have provided an opportunity to confront her knowledge. The geometrical fact of 180 degrees inside a triangle is at the forefront of her thinking when she explained the challenge of drawing an equilateral triangle and was reinforced when the pupils provided the same explanation. Sally did not use Scratch live in this episode which may also have thrown light on this misconception.
In Sally’s post-lesson interview [ESE1(Int)], she explained her intention for the activity:

I expected that they would think that the square was the least challenging. The triangle I would actually put as number 3, 4 or 5 [most challenging] depending on their knowledge. The triangle depends on how confident in their angles in Scratch. The others are all 360, whereas the triangle is 180. [ESE1(Int)]

Sally’s explanation confirmed the exterior angle misconception and suggested that she might consider that bigger numbers (as in the number of sides) make the shape more challenging to draw. This is an example of another misconception since the structure of the polygon is the key idea. The discussion also confirmed and that she was unaware of the *SM Teacher guide* when I used Figure 7.5 from the guide to explain the exterior angle turned through.
7.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

In this section, I present two themes which emerged from the analysis across the teacher data. Within each theme, I will illustrate how Sally characterises the theme and explain the relationship between the mathematical idea and the programmatic representation (see Table 4.5 on page 122 for classification of each theme).

7.3.1 Thinking with your maths head on

The first task in investigation two of the Fireworks microworld has an objective to build and explore different sized squares as a foundation to explore mathematical similarity. Sally introduced Activity 5.2.1 as shown in Figure 7.6 in lesson [ESE(3(Obsv)]. In the task the children were asked to set side length to variable different values, run the square algorithm, and explore the output on the screen.
Sally carefully explained how the **set side length** block could be used to change the value stored in the **side length** variable block. She demonstrated changing the **side length** variable and running the square algorithm, resulted in a square drawn on the screen with the **side length** that had been set. She drew attention to the **move** block inside the algorithm and highlighted that the **move side length steps** controlled the size of the square. The children were then given time to **explore** the blocks and to make the pattern as shown on the below. The consequent task, illustrated in this dialogue, indicated how Sally is beginning to ask children questions that encourage reasoned explanations. In Sally’s previous lessons, her questions were generally more closed e.g. “What block do I need?”, “What does the orange colour mean?”, “Where do I put the variable?”. The questions Sally asked in this episode, are focused on a specific block and require a reasoned explanation from the children.

**SE**: I’d like you to drag in a change side length by.

* [She does this on the IWB].

**SE**: Talk to the person next to you, what do you think this new block might do?

* [The children discuss for 30 seconds].

**Child**: It might change the length of the sides.

**SE**: By how many?

**Child**: Depending on how many, so 1.
7.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

SE: OK. Because at the moment that’s what it says [she points to the change side length by 1 block].

SE: Does it always have to be 1, Tamima?
Tamima: No.
SE: Why not?
Tamima: Because you can change it.
SE: Because it is a? What type of block is it?
Tamima: Variable.

Although Sally has conflated the change block, which operates on a variable with the variable itself, the nature of her questions indicate that she is encouraging the children to explain and reason mathematically. In the post-lesson interview Sally explained how her approach to teaching the children was changing:

I understand what the outcome is more, and because I have a better understanding of how to get there, therefore I teach it better because I know where I’m aiming for. [ESE2(Int)]

She explained that she viewed her developing knowledge as a series of steppingstones and stressed that the space between the steppingstones was just as important as the stones themselves. For example, Sally explained that she understood how the set and change blocks affected the variable (the steppingstones), and because of this understanding she was able to ask the children further questions to connect (the space between) how the value of the variable impacts the output of the square algorithm. She explained, “[They] know what they are in isolation…the steppingstones are there, but the bits between the steppingstones just weren’t.”.

Sally encouraged the children’s explanations of setting and changing variables through her exploration of the set variable to and change variable by blocks in the subsequent lesson [ESE4(Obsv)]. She started the lesson with a review of the previous lesson and her screen was set up as shown in Figure 7.7 below. Each script sets side length to a specific value 30, 40, 50 or 60, and then uses that value to draw a square. The children were asked to explore the output of each of the isolated scripts as shown on the right-hand side of the figure and tasked with creating an explanation of what the set side length block does. Sally asked the children to explain:
SE: On your screens when you press that script, what happens when you press set side length to? Aeris?

Aeris: It just puts a square that gets bigger each time.

SE: Who agrees with Aeris that it makes the square bigger?

[Silence.]

SE: I’m asked you, purely on its own, what does that block do? Do you want to keep your answer or change it?

Aeris: Change it. Because errr.

SE: I asked you if you wanted to keep your answer or change it. You originally said the square gets bigger and that set side length makes the square bigger. You said you want to change it, why?

Aeris: [Inaudible]

SE: Aeris is now saying, he thinks that it is something to do with the line.

SE: What do we mean by it is something to do with the line?

SE: There are some clues, some words in that block that might help you with this. Mahira?

Mahira: It’s the side length of where the sprite is, and it is changing the side length to the number.

SE: Excellent. What is in maths, if we were to think of a way we might describe side length instead of using side length? What would we call it?

[Silence]

Sarah: Side?

SE: Good, what else?

Child: Edge.
7.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

[She writes the words on the flip chart]

SE: So, even though when the script would make a bigger square [she points to the square definition], the actual set side length does what? Tell your shoulder partner now, off you go.

[The children talk with each other]

SE: I’ve heard a couple of different things: it actually states how long the side length is going to be, I’ve heard a few people say it makes it longer. It will only make it longer if your number is then bigger than what you have already got there, it will only make it shorter if the number you are putting in is smaller than the number that’s already in there.

The lesson episodes described in this section demonstrate how Sally effectively built a bridge from the computing to the mathematical context through her use of questions to encourage the children to explain their thinking. She used a metaphor of steppingstones to be to explain how the ideas linked together and her role as a teacher to create the links between the ideas. For example, by creating a link between the programming representation of variable used in the square algorithm and the square drawn as a mathematical representation on the flip chart, she was able to support the children to consider the effect of the changing the variable side length. She asked the children to consider the **side length** block specifically in the maths context “What is it in Maths?”, an example of thinking with your marks head on, so that the children could see how changing the value of the variable in Scratch, changed the length of the side of the square that was output.

7.3.2 Thinking like a sprite

In this section I present Sally’s use of pedagogical strategies which exemplify the Thinking like a sprite them which arose from comparative analysis across the data.

Sally did not have first-hand experience of playing beetle because she had missed the PD due to starting at Emerald at the beginning of Year 6. and as discussed in section 7.2.2 had missed the key idea of beetle turn. However, after engaging with the *SM Teacher guide* Sally created a cut-out beetle to support the children with their understanding of the beetle’s movement and turn. (Playing beetle is encouraged in the Year 5 curriculum materials and is a key aspect of envisage of the 5E pedagogical formwork within the PD).
7.3.2.1 Thinking like a Beetle for variable

In this teaching episode from [ESE4(Obsv)], Sally used the Sequence of Squares activity (see Figure 7.8 below, and for further detail see section 4.2.2.1 of Chapter 4.)

Several children connected all the three blocks together as shown in Figure 7.9 and were surprised by the lack of a pattern other than a square each time they clicked the blocks.

Connecting all the isolated blocks together in Scratch is a common children’s strategy when solving a task. Sometimes this can result in success for the child. Although they would likely miss the purpose of a task if they were left with no teacher intervention and did not have an opportunity to reflect on their outcome. However, in the lesson Sally used the child’s strategy
as an opportunity to debug the script and to develop the children’s reasoning skills, by using the *thinking like a sprite* simile. The following classroom dialogue illustrated how Sally did this:

**SE:** Let’s imagine the beetle. What are we telling the beetle to set the side length to?

**Child:** 50.

**SE:** So therefore, I’m going to draw a side length of 50. I go up and stop there, that’s my 50. I’m then. Hang on I’ve said it wrong, let me start again. I’ve set my side length to 50. He knows my side length is?

**Child:** 50

**SE:** What am I now being asked to do in my script?

**Child:** Change side length [*several in unison*].

**SE:** No, I’m not, not yet I’m not, Fahim?

**Fahim:** Make a square.

**SE:** Good, what am I now going to do?

**Child:** Make a square.

**SE:** So I’m going to draw my square, 50, turn, 50, turn, 50 turn, [*She demonstrates the movement using her cut out beetle as shown in Figure 7.10*]

![Figure 7.10 Sally using a cut out beetle to represent the beetle's movement as it draws a square](image)

**SE:** Now what does my script ask my beetle to do? Have a look, it’s on the board Udjin?

**Udjin:** Change the side length by 10!

**SE:** So now the beetle is going to change my side length to what Sabrina?

**Sabrina:** [Pause]
SE: Is it 10?
SE: We just said that it changes by 10, but then what happens. We’ve gone around, what side length was it?
Sabrina: 60?
SE: What do you mean by 60?
Sabrina: [Pause]
SE: Who can help her? Emtiaz?
Emtiaz: Basically, you add 10 more steps onto the old side length.
SE: What does my beetle now have saved as the side length?
Sabina: 50?
SE: Put your hand up if you think 50?
SE: Put you hand up if you think 60?
[Lilly puts up her hand to indicate 60]
SE: Why 60, Lilly?
Lily: Because you are adding 10.

The episode illustrated Sally was able to think on her feet as she caught herself making a mistake at the start with her explanation. She explained that the set side length block also drew the square, a possible misconception since the square is only drawn when the block is run. Sally was careful to be precise with her language as she explained each step. She used the beetle as an object to think with when she asked “Let’s imagine the beetle. What are we telling the beetle to set side length to?” Thinking about the beetle in this way, provides an effective way to identify firstly what the variable is, e.g., side length. A connection can then be made between the value and what is stored by the variable, e.g. the beetle sets side length to a value and stores it. The next block processed changes the side length by 10. Sally asked “So now the beetle is going to change side length to what Sabrina?” connecting the Scratch programming language representation with the beetle who increased the value of the variable by 10 and stored it. Sally confirmed this by asking “What does my beetle now have saved as the side length?” Of course, the beetle is just a sprite, a particular organisation of pixels that take the form of a beetle, but to the children appears to be an effective bridge to help them to make connections between variable in computing, and a variable in mathematics representing the side length of the square.
7.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

7.3.2.2 Thinking like a Beetle for angle

Sally’s decision to use the beetle was when she had noticed in the previous lesson that many of the children were struggling to explain how to draw a square when a variable was used within the script (Figure 7.11 below). In lesson [ERE4(Osv)] Sally moved the cut-out-beetle on the flip chart as she read out each block of the script in sequence.

![Defined square block using side length variable](image)

Figure 7.11 Defined square block using side length variable

At the end of each step, she asked the children what would happen. For example, after the beetle had moved 4 steps along the side length, she asked what the beetle would do next. A child responded that the beetle turned 90 degrees. Sally then carefully demonstrated this instruction by rotating the beetle. She continued to describe and ask for an explanation until the beetle had moved around the square. Later in the lesson, Sally used the beetle cut out to explain what the beetle was thinking when the side length variable is changed using the change by block. She said “Let’s imagine the beetle. What are we telling the beetle to set the side length to?”. Using the beetle as an object to think with appears to have emerged from Sally grappling with the ideas of angle and variable when teaching with Scratch. Sally used the cut-out beetle to show the beetle’s movement and rotation and to show how linking the physical beetle representation with the programming language and encouraging a thinking like a beetle approach provided effective opportunities for the children to make connections between the concept of a variable quantity and operating on a variable such as in the move side length block.
7.4 Teacher pedagogic practices

In this section, I identify and exemplify the kinds of whole-class activity types utilised by Sally as she taught using the fireworks microworld. I then go on to analyse the quality of Sally’s use of orchestrations and the adherence to the ScratchMaths curriculum through use of the adapted TRU framework for teaching mathematics through programming as discussed in section 3.6.4.

7.4.1 Whole-class orchestrations

After the first observed lesson, it became apparent that Sally was unaware of the ScratchMaths teacher guide. Her first observed lesson used pedagogical approaches she had developed from her previous experiences of teaching Scratch instead of the 5E pedagogical framework. For example, Sally used Scratch to demonstrate the final fireworks Scratch project as well as shared the work from two children. There was no discussion, explanation or bridging to mathematics content. Sally’s predominant approach in the lesson was to show and to tell the children what to do. At one point in the lesson, they had to copy the provided final fireworks Scratch script for themselves. In the post-lesson discussion [ESE1(Int)], I explained an overview of the ScratchMaths curriculum, including a description of the 5E pedagogic framework and provided Sally with access to the teacher guide and to the Year 5 materials.

Following the intervention, in every subsequent observed lesson Sally used whole-class orchestrations more frequently. Sally used live Scratch programming for an increasing amount of time across the observed lessons. In the first lesson there was no live programming, but in each following lesson Sally used Scratch more frequently to explore mathematical and computing ideas in mathematics. By the final observed lesson, Sally was using whole-class orchestrations (Discuss-the-screen, Explain-the-screen, Sherpa-at-work) i.e., using Scratch interactively, for most of her whole-class teaching time.

Table 7.2 provides an exemplification of each type of whole-class activity and their frequency observed across the four lessons.
### Table 7.2 Whole-class activity types, their characterisation and frequency observed across the four observed lessons

<table>
<thead>
<tr>
<th>Whole class orchestration type</th>
<th>Characterization of whole-class orchestration</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical-Demo</td>
<td>Sally used a Scratch technical demonstration in each observed lesson. In the first observed lesson there were several examples of technical demonstration. For example, she reminded the children how to access the ScratchMaths curriculum projects and how to save their own copy, as well as describing where to find specific blocks. Sally frequently demonstrated an instruction as she asked the children to perform the same instruction. For example: “I want you now to drag in your set side length to 0 from data, and I then want you to grab a square and pop it underneath. Can we do that now?” [ESE3(Obsv)]</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Discuss-the-screen</td>
<td>The <em>Discuss-the-screen</em> orchestration was used most frequently in the observed lessons. Sally would typically ask the children a question focused on the content displayed on the IWB. The children were given the question to discuss in pairs, or as a question posed to the whole-class. In both situations the children fed back their explanations. In one episode, when Sally taught a smaller group, to ensure participation from all pupils, she asked the question and then asked the children to vote with their eyes closed. She followed up with a non-voting child to ask them to explain what they were thinking.</td>
<td>Frequently</td>
</tr>
</tbody>
</table>
| Explain-the-screen           | Sally used the orchestration infrequently and only in the latter two of the observed lessons. In lesson [ESE3(Obsv)] she linked the programmatic structure of a square to the mathematical structure of the square. In the final observed lesson, three months had passed since the first observed lesson and demonstrated a richer use of the orchestration [ESE4(Obsv)] Sally reflected at the end of Year 6 and the progress she had made, and her hopes for teaching the curriculum for a second time in the upcoming year. She identified that making links to the mathematical content had been missing in her teaching and is an area which she wants to focus on:  

I hope by going back to the beginning, it will help with the box that is kind of missing... in terms of the maths and the links. It’s more the links and the connections, when I think about it, I know, but it’s needing [that time] to think about it. [ESE3(Int)] | Sometimes |
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sherpa-at-work</td>
<td>The <em>Sherpa-at-work</em> orchestration was used in two of the observed lessons. Sally used the orchestration to demonstrate children’s work onto the IWB so that all the children could see the steps, and the final outcome. She built the script live as the child instructed what to do. If the script required user input, such as setting a variable to a specific value and changing the default value, she asked the children exactly what value to type in. The approach to build the script live, as opposed to opening the child’s work, was described by Sally as being more efficient due to the small number of blocks used.</td>
<td>Rarely</td>
</tr>
<tr>
<td>Board-instruction</td>
<td><em>Board-instruction</em> was used in each observed lesson at the start of each lesson to recap previous work. The children had access to their previous lesson’s work and used it to support their explanations. Sally used a flip chart and a digital flip which could be annotated. In the observed lessons Sally did not make use of any paper based unplugged activities away from the computer.</td>
<td>Frequently</td>
</tr>
<tr>
<td>Link-multi-modal-representation</td>
<td>Sally used the <em>Link-multi-modal-representation</em> to connect roleplay with the Scratch language during her first observed lesson [ESE1(Obsv)]. In the example, she acted as the sprite, or playing computer. She followed the instructions provided by a child to make explicit that a script in Scratch will only follow the instructions that you provide it. The detail of the interaction is illustrated in the example below. Sally also used other resources such as a cut out beetle to demonstrate its movement on the flip chart as discussed in section 7.3.2.1 Thinking like a Beetle. She also used multilink cubes to represent the variable in the side length of the square [ESE4(Obsv)], the details of this episode is elaborated in the example below.</td>
<td>Sometimes</td>
</tr>
</tbody>
</table>
A first observation of the table is the frequent use of the *Technical-Demo* orchestration. The result can be partly explained by Sally’s lack of experience with ScratchMaths as she had not taught the curriculum the previous year. She was therefore unaware of what the children had been taught and their Scratch knowledge. Consequently, Sally explained the technical aspects of Scratch to the children as if they were experiencing Scratch for the first time. For example, showing the children where to find blocks, how to save work or how blocks connect. Sally explained in an early post-lesson interview [ESE1(Int)], that she found her existing Scratch knowledge to be lacking when she started ScratchMaths. Her previous experience of Scratch had been working with copying or editing Scratch projects, rather than programming from nothing. The copying and editing strategy were evidenced in Sally’s first observed lesson [ESE1(Obsv)]. However, the ScratchMaths pedagogical approach which encourages exploring and explaining a single block were evidenced in later lessons after Sally’s engagement with the *SM Teacher guide* [ESE2(Obsv)], [ESE3(Obsv)] and [ESE4(Obsv)].

A second observation is the high frequency of the *Discuss-the-screen* orchestration. The orchestration was used by Sally in each observed lesson. The frequency increased in [ESE3(Obsv)] and [ESE4(Obsv)] as Sally spent more time asking focused questions of the children to encourage connections between the Scratch blocks and how they are used within the task context. Sally explained that she used questions to focus on the gaps between the ‘steppingstones’ of the task, if the steppingstones were the smaller objectives along the path to achieve the overall objective (See also section 7.3.1). She accomplished this by breaking a task down into a series of a smaller objectives and then asked the children to explain each of the smaller steps to form connections between the steppingstones.

### 7.4.2 Link-multi-modal-representation orchestration

**Role play to support step by step processing and reasoning of a script**

In this episode [ESE1(Obsv)] Sally taught a lesson from Investigation 1 of Module 5 – Polygon Fireworks. She had displayed on the board the slide shown in the Figure 7.12 below.
Sally asked the children if the solution on the board was correct and to explain their decision. Sally used the *Link-multi-modal-representation* orchestration to link the physical actions of role play or *playing computer* with the Scratch programming language. By linking the representations Sally helped to support two key ideas of algorithms, the sequential processing of the script and the computer will only process the exact instructions provided.

**SE:** Is the solution on the board correct? And why? Emtiaz?

**Emtiaz:** There is one reason for it, I don't think it will work. Because when it says "What the side length?" and "How many sides?", that's fine but they haven't used the block for how many polygons and when they use the script, they won't know how many to draw.

**SE:** OK. Think of it another way. You're a robot and can only do what you've asked me to do.

**SE:** Hamim, come and stand at the front. You're going to be the robot. You're going to ask me, 'Do you want a cup of tea?'

**Hamim:** Do you want a cup of tea?

*[Sally responds with no action, staring blankly.]*

**SE:** What have I just done? Why is Hamim not making me a cup of tea? In my head I've thought it, but I didn't do what? Martin?

**Martin:** You didn't tell him the answer.

**SE:** I didn't tell him the answer.
The use of role play is an additional mode of representation imitates the structure of the programming environment without the use of technology. Sally does not develop the role play situation further to explore the limitation of holding one answer or when the answer is overwritten when a second question is asked. A second example of the link-multi-modal-representation orchestration is exemplified when Sally used multilink cubes to represent the concept of a variable.

**Multilink cubes to represent beetle steps and side length of squares**

Towards the end of the observed lesson [ESE3(Obsv)], the children worked on the Altering polygons: Activity 5.2.1 task (see section 4.2.2.1) to explore the output of small algorithm scripts and engage with the impact of setting and changing variable when using a defined square script. Figure 7.13 shows Sally’s screen.

![Scratch project](image)

Figure 7.13 Scratch project to show isolated scripts which set side length variable to different values and then use it to draw a square

The children had little difficulty trying for themselves to change the values of the variables by clicking the blocks and observing the square pattern outputs since the task is trivial requiring little more than typing and mouse skills. However, the children required considerable scaffolding during Sally’s whole-class discussion task. She asked the children to explain how the value of the variable changed as each script run, and how the value of the variable affected the square drawn. To support the children with their explanations of each step, Sally designed
a task in the subsequent lesson which used a multilink cube to represent the value of the variable. In the image below, Sally is holding a four-by-four multilink square, the computer screen is shown alongside her white board.

Figure 7.14 Sally holding a four by four multilink square to link the representation of a square drawn on the screen, the Scratch script and the drawing on the white board

Sally explained how the multilink cube represented the variable:

**SE**: We started to look last week at 'changing side length'.

**SE**: I'm going to give you some multi-link cubes… First of all, you are going to make a square with your cubes, that is 4 by 4. When I say, 4 by 4 what do I mean? Aeris?

**Aeris**: It has 4 blocks going down, and 4 blocks going across.

**SE**: Excellent, so each side length will be 4 cubes.

*The cubes are given out by the TA and the children make the square using the multilink cubes*.

**SE**: Hands on your heads... If we were to think of this, in terms of setting our side length. What would we put into our box? *She points to the set side length block on the screen* If we were doing it for this square we have in front of us? What would go in our set side length? Adeya?

**Adeya**: 4.

**SE**: Why do you think 4?

**Adeya**: Because there are 4 *inaudible*.

**SE**: What does each of the blocks show?
Adeya: 1 step.

SE: So, if we had the beetle, and we were to do the four … we would start here and we would go, 1, 2, 3, 4. [She moves the beetle the four steps along the side length.]

Figure 7.15 Sally showing the cut out beetle moving the side length as represented on the white board

SE: Then what would we do, Samiyah?

Samiyah: Turn right 90.

SE: Good, turn right 90. Then what would I now do?

Samiyah: Move 4 steps forward.

SE: Excellent, so I would move another 4.

The dialogue demonstrates how Sally has represented the side length variable used within the square algorithm (Figure 7.15) by linking multiple representations of the Scratch language, the physical beetle, diagrams on the white board, and the use of multilink cubes. Sally ensured that the child was able to explain that each cube represented 1 step that the beetle moved. Through her use of the cut-out beetle, she was able to demonstrate the beetle moving 4 steps along the side of the square. This would be challenging to show using Scratch since move 4, (as represented by the multi-link square) would move 4 pixels, a very small amount of movement on the screen, and be largely undetectable. The representations helped to support what is a challenging explanation. A full explanation requires the child to perform multiple and sequential steps in their reasoning i.e., to hold the value of the variable concurrently as thinking about how the move block operates on the variable within the definition of the square. However, by linking together the different modes of representation, Sally was able to make a connection between the value of the variable (the four cubes), and the variable operated on in the move block as part of the square algorithm.
In this section I report on the analysis of Sally’s observed lessons through use of the adapted TRU framework. The framework’s numerical scores represent a Basic (1), Proficient (2) and Distinguished (3) performance in each dimension (see Appendix A for the full scoring rubric). The results for each observed lesson are shown in Table 6.3 and represented as a chart in Figure 6.26 below.

<table>
<thead>
<tr>
<th>Date</th>
<th>Microworld</th>
<th>Mathematics through programming</th>
<th>Cognitive demand</th>
<th>Equitable access to content</th>
<th>Agency, Ownership and Identity</th>
<th>Formative assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>22/03/2017</td>
<td>Fireworks</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>05/06/2017</td>
<td>1.5</td>
<td>1.5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21/06/2017</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>28/06/2017</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 7.3 Weighted averages for each dimension of Sally’s observed fireworks microworld lessons using the adapted TRU framework for Maths through programming

**Figure 7.16 Chart to show weighted averages for each dimension of Sally's observed lessons using the adapted TRU framework for Maths through programming**

**Dimension 1, Mathematics through programming**

Sally’s use of the 5 E pedagogical framework developed over the three month period she was observed. In the first observed lesson [ESE1(Obsv)], before Sally was aware of the framework, she provided some opportunities for the children to explain their thinking and to explore Scratch projects using the computer. Her responses to the children’s explanations were limited and did not suggest connections with the mathematical objective of the task or support the
children to develop their thinking further. However, in the latter lessons, Sally changed her teaching approach by breaking down each task into a series of smaller steps as suggested in the *ScratchMaths teacher guide*. The steps provided Sally with an opportunity to create focused questions to ask the children and provide rich opportunities to *explain*. For example, in this extract [ESE4(Obsv)] Sally used one Scratch script to draw two different sized squares. (The two squares are different sizes because the variable changed its value before the script was re-run). Sally asked the children to *explain*:

**SE**: Talk to the person sat next to you, why are my squares different sizes? Off you go.

*The children discuss for 30 seconds*.

**SE**: Rydda, have you got an answer for me?

**Rydda**: So you know the number for the side length, whatever number you put in is the moves. The lower the number is, the smaller the square is. The bigger the number is, the bigger the square is.

**SE**: Wow! OK I love that answer… 'The number that you put in the box is the amount of moves that your beetle will make, and the bigger the number, the longer the side length will be.'

Sally did not use any paper based unplugged activities which can provide rich opportunities to *envisage* a script’s output in the observed lessons. However, in [ESE3(Obsv)] and [ESE4(Obsv)] she encouraged the children to predict what the output might be when she was coding live, by asking “what would happen if?” questions.

Sally’s use of *bridge* was limited in the first three observed lessons and connections made to mathematics were brief, for example in [ESE1(Obsv)] as discussed in Section 1.3.1 above. In the episode Sally was unclear on the external angle the beetle turned through and provided a muddled explanation of external and internal total, in the context of explaining how to calculate the external angle of turn for a regular polygon. However, the final observed lesson demonstrated Sally making greater links to mathematics, and more focussed bridging, as illustrated in Sally’s use of a cut out beetle to correctly show its turn, and the use of multilink cubes to support the explanation of variable in a square algorithm. Sally reflected in her post-lesson interview [ESE3(Int)] that she will make greater links to mathematics when she teaches the curriculum again. She recognised that these were frequently missing since she was grappling with unfamiliar content at the same time as developing unfamiliar pedagogical approaches.
From the scoring rubric, Proficient: The teacher uses orchestrations and engages with the 5E pedagogical framework but connections with mathematics (bridgE) are limited. Students are given some opportunities to Explore, Explain, Envisage, and Exchange when working with the computer.

**Dimension 2, Cognitive demand**

Sally used the ScratchMaths materials in each lesson following the first observed lesson when it was identified that she was unaware of the *SM Teacher guide*. Sally explained that her planning process was to read the teacher materials before each lesson and to try and build each of the Scratch projects for herself. Sally had not attended the PD for Year 5 or Year 6 as she was new to the school, so she had bought and was working through a Teach Yourself Scratch book. She recognised that the approach to ScratchMaths activities was different than other Scratch curriculums she had used at her previous school as other curriculum just require children to copy scripts. She used the Teach yourself Scratch book to provide her with deeper explanations of the different Scratch blocks and structures. Sally did not adapt the ScratchMaths activities to provide additional opportunities for the children to reason mathematically or computationally, i.e., she did not create her own projects for the class to debug or use children’s projects as opportunities to debug. In her first observed lesson, she did create her own task to discuss the complexity of drawing regular polygons, however, as discussed in section 7.2.2 there were mathematical errors in her teaching and a lack of coherence in the explanation.

From the scoring rubric, Proficient: Classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to “scaffold away” the challenges, removing opportunities for productive struggle.

**Dimension 3, Equitable access to content**

Sally’s first observed lesson [ESE1(Obsv)] had uneven child engagement as they worked on the computer tasks. This was largely a consequence of Sally being unaware of the *ScratchMaths teacher guide* and being unable to fully explain the purpose of each task [ESE1(Int)]. Her objective for the lesson was to create polygon fireworks but had not broken down the task into manageable pieces. Her approach was for the children to recreate her working polygon fireworks, by copying it block by block. They were also given the option of
creating the final project themselves without copying. However, many children tried to do it for themselves and became stuck and asked for help. Whilst the children waited for Sally, they went off task and played with the different sprite graphical options. However, Sally is an experienced teacher with well-developed classroom management strategies as reflected in latter lessons when Sally used the ScratchMaths tasks and guidance. Despite arriving new to the school, in the previous term, Sally reflected that her pedagogical practices to encourage collaboration between children, such as “talk to your shoulder partner” (meaning the child sat next to you), or “help your shoulder partner” (to encourage explanation and collaboration), were approaches that the children had experience of and were expected to engage with at the school [ERE5(Int)].

From the rubric: *The teacher actively supports and to some degree achieves broad and meaningful mathematical participation.*

**Dimension 4, Agency, Ownership and Identity**

The amount of agency a student had or developed in Sally’s classroom seems to be largely related to their participation. Sally called upon students either by name directly or responded to those who raise their hand first. In the first three observed lessons, Sally maintained strict control over the direction of the conversation in the classroom, always initiating conversation through her questions. The final observed lesson indicated a loosening of the control as Sally provided more opportunities for the children to extend their explanations rather than accepting short one sentence or less explanations. This suggested Sally was developing confidence with teaching mathematics through programming. From the scoring rubric the form of classroom discourse lies somewhere between *Basic: The teacher initiates conversations. Students’ speech turns are short (one sentence or less), and constrained by what the teachers says or does and Proficient: Students have a chance to explain some of their thinking, but the teacher is the primary driver of conversations and arbiter of correctness. In class discussions, student ideas are not explored or built upon.*

**Dimension 5, Formative Assessment**

The development of formative assessment in Sally’s lessons closely followed the development of children’s agency and ownership as in the previous dimension. In the first three observed
lessons, children’s explanations are mostly composed of descriptions of that they did, or what they would do, e.g. “What and How do you?” type questions rather than “Why?” questions. Sally responded to children’s responses with encouragement and did not unpick their reasoning. There was evidence of this beginning to change in the fourth observed lesson as Sally started to introduce a follow up “Why?” question to children’s responses. Sally confirmed this observation of her practice when she reviewed the video footage of the lesson a year later:

I’m isolating the ‘what’ of what I’m talking about. I’m more confident, you can see it in how I’m communicating with the children, I’m direct. You can see that I’m not lingering as much.

[SSE5(Int)]

She went on to say how she was less prescriptive in this lesson than in the earlier lessons as she provided more opportunities for the children to explore and explain their thinking. Sally felt that this was a result of her having a better understanding of what she was teaching because of teaching it, and because of engaging with the SM Teacher guide.

From the scoring rubric Sally’s use of formative assessment lies somewhere between Basic: Student reasoning is not actively surfaced or pursued. Teacher actions are limited to corrective feedback or encouragement and Proficient: The teacher refers to student thinking, but specific student’ ideas are not built on (when potentially valuable).

7.5 Summary

In this chapter, I have presented an analysis of Sally’s case data organised as i) windows on mathematical knowledge, ii) mathematical ideas mediated and mediated by the ScratchMaths curriculum, and iii) teacher pedagogic practices, to examine her mathematical pedagogical technology knowledge (MPTK). The analysis of Sally’s MPTK provided evidence to illustrate how her knowledge has developed through teaching ScratchMaths. Sally had existing knowledge of variable in computing from her experience prior to teaching SM. She defined a variable as data which can only hold one value at a time and that the value can be changed depending on when certain conditions are met. However, as illustrated in the early teaching observation, she found it challenging to explain the concept coherently to the children when explaining multiple variables within a regular polygon algorithm. She stated facts that she knew about variables in the context of Scratch and was unable to provide a coherent explanation as to how they were being used in the algorithm. However, in later observed
7.5 Summary

lessons, Sally used Scratch live to set, change and operate on variables in multiple contexts including mathematical contexts such as the general algorithm for a regular polygon. She demonstrated that linking together multiple representations created an effective bridge to create connections between expressions in Scratch and in mathematics, for example, using and operating on the side length variable in the square algorithms. Analysis of Sally’s MPTK suggests that she was beginning to form a stronger link between the concept of variable in computing and variable in algebra.

Misconceptions in Sally’s geometric knowledge were surfaced as she taught with the Scratch microworld. She had missed that the beetle turned through the external angle and used disconnected maths angle facts to explain the angle of turn in a regular polygon. The misconceptions were discussed in an interview intervention. Sally’s creation of a cut out beetle to teach the children indicated that she had resolved the misconception. The angle of turn in a regular polygon was returned to in an interview based task towards the end of the data collection and provided further evidence of the reconciled misconception.

Sally’s pedagogical approach during her whole-class teaching developed over the year. In the early lessons she mostly told the children what to do and set large objectives for the children to achieve when working on the computer. However, in the latter observations she broke down large objectives into much small objectives creating a sequence of steps which would address the gaps between the bigger steppingstones of the task. She encouraged the children to explain each step of an algorithm and to make connections between the steps. The data illustrated Sally using the beetle as an object to think with in the contexts of variable and angle. She used a cut-out beetle on the flipchart, as well as used the beetle metaphor to ask what value was held to support the development of the variable concept. Sally used multilink cubes, a cut out beetle, and role play, all different modes of representation which helped to exemplify the classification of another type of whole-class orchestration when teaching with Scratch to teach mathematical concepts, Link-multi-modal-representation.

Analysis of the lesson observations using an adapted TRU framework for mathematics through programming indicated that Sally was using the ScratchMaths investigations as designed after the initial lesson (Dimension 2, Cognitive demand). However, she was not yet ready to adapt the materials and create her own scripts to debug and provide the children with additional
opportunities to build understanding and engage in mathematical practices. Sally developed her use of the 5 E Pedagogical framework across the observed lessons, although links to mathematics, bridge and envisage for script output were limited. However, her final observed lesson demonstrated engagement with all of the 5 E’s. Sally had recognised that the mathematical links were not well established and would become possible as her confidence grew with the curriculum materials (Dimension 1 - Mathematics through programming).

After Sally became aware of the ScratchMaths teacher guide, Sally was able to create a classroom with high levels of meaningful mathematical participation from all students (Dimension 3 – Equitable access to content) through her use of established routines and structures. Children in Sally’s classroom were provided with some opportunities to explain their reasoning although Sally had generally strict control over the initiation and direction of classroom discussion. (Dimension 4 – Agency, Ownership and Identity). In the earlier lessons she tended to select the children who were more confident with programming. However, Sally’s final observed lesson demonstrated a developing confidence with the ScratchMaths curriculum and showed a greater inclusion as she provided support to the weakest children. The children also had more opportunities to explain their thinking, and featured classroom dialogue which developed their answers more fully (Dimension 5 – Formative Assessment).

Table 7.4 below provides a summary of the findings and their location within each section of the chapter where they are exemplified.
Table 7.4 Table to show summary of findings and location within the chapter for Sally’s case study

<table>
<thead>
<tr>
<th>Window on mathematical knowledge for teaching</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Window on variable</td>
<td></td>
</tr>
<tr>
<td>• Used Scratch programming to help make sense of variable concept in computing and in mathematics.</td>
<td>7.2.1</td>
</tr>
<tr>
<td>• Created knowledge of the limitation of the answer block in Scratch and using variables to overcome it.</td>
<td></td>
</tr>
<tr>
<td>• Developed connections between computational representation and algebraic representation of variable.</td>
<td></td>
</tr>
<tr>
<td>Window on angle</td>
<td>7.2.2</td>
</tr>
<tr>
<td>• Use multiple representations to link geometric ideas in programming, in mathematics, on screen and on paper.</td>
<td></td>
</tr>
<tr>
<td>• Developed connections between geometric structure in mathematics and in programming.</td>
<td></td>
</tr>
<tr>
<td>• Teaching using the fireworks microworld unearthed geometric misconceptions of drawn angle.</td>
<td></td>
</tr>
<tr>
<td>How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum</td>
<td>7.3</td>
</tr>
<tr>
<td>Thinking with your maths head on</td>
<td>7.3.1</td>
</tr>
<tr>
<td>• Used Explain to interpret process of scripts step by step to encourage mathematical reasoning and make connections between programming and geometric representation.</td>
<td></td>
</tr>
<tr>
<td>Thinking like a sprite</td>
<td>7.3.2.1 7.3.2.2</td>
</tr>
<tr>
<td>• Used the beetle as an object to think with in angle and variable contexts.</td>
<td></td>
</tr>
<tr>
<td>Teacher pedagogic practices</td>
<td>7.4</td>
</tr>
<tr>
<td>• Whole class orchestrations</td>
<td>7.4.1   7.4.2</td>
</tr>
<tr>
<td>• Use Scratch live programming for an increasing proportion of each lesson.</td>
<td></td>
</tr>
<tr>
<td>• Increasingly used Link-multi-modal representation, included role play, a cut out beetle and multilink cubes to represent programming ideas.</td>
<td></td>
</tr>
<tr>
<td>• Frequently used Discuss-the-screen orchestration</td>
<td></td>
</tr>
<tr>
<td>• Frequently used Explore and Explain. Bridge was most effective when using and linking multiple representations. [D1]</td>
<td></td>
</tr>
<tr>
<td>• Used SM curriculum materials more consistently after the initial observed lesson [D2].</td>
<td></td>
</tr>
<tr>
<td>• Developed more consistent classroom routines for children and teacher to engage with each other [D3].</td>
<td></td>
</tr>
<tr>
<td>• Demonstrated a loosening of control and invited more children to explain their thinking as her confidence grew. Used more why type questions in later lessons to surface student thinking [D4, D5].</td>
<td>7.4.3</td>
</tr>
</tbody>
</table>
Chapter 8  Teacher episodes

8.1  Introduction

In this chapter, I describe and analyse data collected from teacher episodes and interview tasks for teachers who were observed once over the course of data collection. See section 3.3.1 for teacher profiles and background. They are an important contribution to the data collected since they represent a wider sample of teachers beyond the three cases and can provide additional exemplification of teachers’ evolving MPTK as they teach with the ScratchMaths curriculum materials. The chapter is structured into three sections i) Windows on mathematical knowledge for teaching, ii) Technology shaping and shaped by the mathematics, and iii) Teacher pedagogic practices. The relationship of the three sections to the research questions and mapped on to the conceptual framework is shown in section 4.3 of Chapter 4.

8.2  Window on mathematical knowledge for teaching

In this section I describe and analyse selected observations of teaching episodes for Group A and Group B’s teaching with Place value and Fireworks microworlds. I also discuss and analyse the teachers’ responses to the task-based interviews. Both approaches opened a window onto their mathematical pedagogical technology knowledge.

8.2.1  Window on place value

Group A

From Group A only Priya and Anthony taught using the Place value microworld and consequently responded to the place value MKT task (see section 3.4.2.1). Anthony’s answers
to each question identified the error and gave limited explanation of the error. For example, his answer to Question 1:

The 31.3 they put it in the wrong place, they must think it's three hundred and thirteen. So, they've ordered it wrongly. I think that's it. [FAO(PVT-Q1)]

Whereas his year 6 partner teacher, Priya, described the error but also provided an elaborated explanation to focus on what the child could do to solve the problem:

They need to look at the thirty-one, which is a tens and ones, so that needs to go before. So, the same with that one that’s nought point, so that needs to go at the beginning. [FPR(PVT-Q1)]

She explained that she might use place value charts to support the children with their understanding of the value of each place. A difference in the two teachers’ mathematical knowledge for teaching is highlighted in their response to the second question. Anthony correctly identified the errors in each of the two-digit addition problems, but his mathematical language was imprecise, “They've carried a 1 instead of a 2.” [FAO(PVT-Q2)] and “That should be a thirteen, basic addition mistakes”. He did not use the language of one ten, or two tens in his explanation of the carry/exchange. However, Priya’s answer demonstrated a deeper understanding of the misconception:

[They] Probably don’t understand the concept, they must have worked that out as 22, but they probably don’t understand the concept of how many tens, they probably think that one ten, put one over, but it’s actually two tens. [FPR(PVT-Q2)]

Anthony’s answer to the third question indicated the same surface level of knowledge with little reasoning or explanation: “The child has just taken it [the 2 in 22] for individual numbers rather than place value.” [FAO(PVT-Q3)]. Whereas Priya’s response is more nuanced, suggesting a deeper understanding of the child’s misconception.

So, she had not understood that it is two tens, she just thinks that it is two and three … She’s not understanding the representation of this as being two tens, just thinks of it as 2. [FPR(PVT-Q3)]

Although the context of explaining in an interview setting is different than classroom teaching, Anthony’s responses suggested a weakness in his mathematical knowledge for teaching place value, particularly in comparison to his partner Year 6 teacher Priya’s. Priya’s responses demonstrated a mathematically precise language and drew attention to the mathematical structure of the place value system.
Group B

From Group B, Katherine, Laura and Matt taught the place value microworld and responded to the place value MKT task. All three teachers identified the error in Question 1. Laura provided an additional explanation to the error and drew attention to the purpose of the decimal point:

Not understood that the decimal point alters the place value. They are looking at the digits and assuming that they are all part of the same whole number. [ALH(PVT-Q2)]

Katherine and Matt explained that the children had failed to recognise the decimal point and counted the number of digits to compare the quantities. All three teachers identified the calculation error in the algorithms for adding together two two-digit numbers. Matt’s response described the error, e.g., “Here they've carried 1 over, so this isn't correct either, this should be 21 (so it should be a 2 and 1” [BMS(PVT-Q2)]” rather than explaining what the error indicates in the child’s understanding. However, both Katherine and Laura, explained that the third example did not require the children to carry more than 10, so there was not enough data to select which of the three children had the most solid understanding of place value. For example, Laura explained:

The first two it looks like they are only ever moving 1 ten across into the tens rather than two. But the last one is a different sort of mistake, the adding of the digits is the problem. This child appears to be able to do the actual single digit addition but are not writing out their answer correctly.

[Child] C understood the concept better, the place value, but it's hard to tell since the question doesn't require them to carry more than one 10. [ALH(PVT-Q3)]

All three teachers answered the final question correctly and provided a similar explanation to Katherine, “OK this is a really simple place value issue, she is seeing the 2 as two and not 20.” [CKW(PVT-Q3)]. However, only Katherine explained that this error tended to occur when she used counters and therefore preferred to use Diene’s blocks.

Laura shared a concern about her class and explained how teaching the place value model in Scratch had helped raise children’s misconceptions that she had previously missed when teaching the addition algorithm:
Some of the children had no idea why they were moving the numbers across, it’s just because they’ve been told to do it, or they know that’s that you are meant to do. [ALH(PVT-Q10)].

She explained that in her experience, some children had learned the pencil and paper algorithm without understanding the structure of the place value system. She gave an illustrative example of how the Scratch place value representation could support those children, e.g., a child adding on 0.1 to 12.9 and giving the answer of 12.10. She said that this can be quite common, but using the model built in Scratch to consider each place and asking what happens when you get to 9 and add 1 more could support the child in understanding how the places are connected. Although this could be considered as analogous to place value charts, or using practical items such as Dienes blocks, an advantage of the model expressed in programming is that base ten place value models operate with the same rules. Laura explained this in her own words:

You get to a certain point and a message needs to be send to the next column, it’s just at what point, the message needs to be sent that’s different. [ALH(PVT-Q6)].

Matt was similarly asked to think about the relationship of the place value microworld and place value in the classroom. He explained that when he thinks about place value, the written algorithm is predominant e.g., the carrying mechanism in column addition or mentally partitioning a number into its components. However, after teaching with the place value microworld, Matt could see how such a model can be used as an additional resource to support his teaching of place value within his maths lessons:

It would present another tool for me to use, as an aid… I don’t think it’s changed my understanding of place value but added to it, added another opportunity for application and demonstration. [BMS(PVT-Q9)]

He explained that when thinking about the purpose of the ScratchMaths place value microworld, he considered the message sent to the next sprite is the same as the message that is being sent when you are doing addition or subtraction but that the “method of message delivery” is different.
8.2 Window on mathematical knowledge for teaching

8.2.2 Window on variable

Group A

Group A teachers had limited previous Scratch experience or ScratchMaths PD. Betty, Trevor and Elizabeth all taught lessons using the fireworks microworld.

Betty’s observed introductory lesson [HBK1(Obsv)] of polygon fireworks provided a small insight into her understanding of variable in Scratch and in mathematics. She used Investigation 1 of Module 5 as shown in Figure 8.1 below to plan the lesson, although she did not display the ScratchMaths PowerPoint during the lesson. At the start of the lesson, she used the default Scratch stage with the cat because she was not aware of the ScratchMaths starter projects which set up the Fireworks microworld using a beetle sprite.

![MODULE 5: INVESTIGATION 1](image)

**MODULE 5: INVESTIGATION 1**
Activity 5.1.1 – Ask and Answer

- Build a script for the **Beetle** sprite to ask for your name.
  - ask: What's your name? and wait
  - name: Alanna

- Next get the **Beetle** to greet you by name after it has asked what your name is.
  - say: Hello! for 2 secs
  - answer: Piers

Figure 8.1 SM Activity 5.1.1 - Introducing the Ask and Answer blocks.

She encouraged the children to explore the **ask** and **answer** blocks, so that the cat would say the child’s name. Very few children (3/25) completed the task to operate on the **answer** block using the **say** block. E.g., the cat displayed the child’s name on the screen (Figure 8.2) but they had not engaged with the concept of a variable input. The cat would always say Piers whatever was input.
Betty showed a correct solution to the children in the post activity discussion, but their attention was not drawn to the potential misconception of not using `answer` at all. Betty’s approach to the discussion, and not noticing the children’s errors, suggests a weakness in Betty’s understanding of the purpose of the task and of her understanding of how the `answer` block can be operated upon in Scratch. The rest of the lesson veered further and further away from the ScratchMaths material sequencing. For example, Betty encouraged the children to develop questions and answers to simple mathematics calculations and introduced the control `if, then` structure (Figure 8.3).

After this example, she developed the control structure further to use the `if, then, else` structure (Figure 8.4)
The children had some success with the tasks and were able to replicate Betty’s example using their own simple calculations. However, the task required the children to do little more than change the question text and perform a simple calculation. The children are therefore exploring and engaging with control structures i.e., if, then, else rather than operating on answer in different contexts as was intended by the ScratchMaths materials. Betty’s lack of engagement with answer as a variable was confirmed in the final part of the lesson when she asked the children to draw regular polygons using different number of sides. She discussed that the angle of turn for an octagon could be calculated by dividing three hundred and sixty by eight. However, there was no discussion or explanation of the generality of this statement. For example, stating that three and sixty degrees is the total turn for the sprite moving around a closed polygon shape. Consequently, she did not discuss how the answer block could be used to control an aspect of a regular polygon, and the children just created examples of different polygons. A task that most were able to accomplish due to the experiences of a similar task in Year 5.

Betty’s post-lesson interview revealed some of the factors that led to her chosen teaching approach within the lesson. She had been appointed at Christmas, a month before the observed lesson, and had no handover with the teacher that had left the school. Although she worked alongside her partner teacher, she explained that her partner teacher had taught the previous module in the first month, and she had not taught with Scratch in her previous teaching experience. She shared that when she was planning the lesson, she found it difficult to connect
the ScratchMaths teacher guide with the PowerPoint slides and she was not aware of the starter projects: [HBK1(Int)].

**BK:** I had a look at all the notes, and to be honest didn’t make sense to me.

**PS:** Why was that, can you show me?

**BK:** I have this and this ([She points to a printout of the teacher guide and the PowerPoint slides]). So, I decided that from knowing that a lot of the kids haven’t done much at all that the biggest things were ask and answer.

**PS:** When you say, they’ve not done much at all, where is that coming from?

**BK:** I think I was guessing to be honest, and since I’ve been here, I’ve not seen them do much.

**PS:** You had a look at the materials, tell me what you did next.

**BK:** When I fiddled with ask and answer, there is so much that you can do with it, that I thought they needed to get an idea of that first and then the shapes as well.

Betty’s lesson observation highlights the dangers of implementing the ScratchMaths curriculum, with no previous PD and little in school support. Although she had tried to teach using the materials, as was expected by the intervention, she missed key aspects of Beetle Geometry such as the notion of the Beetle’s turn and the purpose of the sprite viewed from above. Without the Scratch starter projects, and the teacher guide, she was unable to engage deeply with the mathematical and computational ideals that had been embedded within the microworld. Although some children did have some limited success with the task, without the teacher intervention and discussion it is unclear whether they were able to bridge to the mathematical context on their own.

Elizabeth, and Trevor were in a similar position to Betty in that they too had inherited the ScratchMaths curriculum at the start of the school year, also due to staff changes, and had not attended any professional development, or received any significant support. I had intended to observe Elizabeth and Trevor over a period of a week as part of a case study. However, on day one, it became apparent in the first observed lessons that despite using the PowerPoint slides, the teachers were not able to support the children with their programming to make progress. They were able to tell the children what to do, and to copy working scripts, but were unable to support to children to debug their work, beyond telling them what a completed script should look. They were not able to explain unexpected output of the Beetle, for example when the Beetle went off the screen. Unsurprisingly, both teachers grew increasingly frustrated teaching
within the lesson. However, I observed that some children were still able to reach successful outcomes to the tasks that they worked on. I suggest that contributing to the successful outcome was the children’s ScratchMaths knowledge developed in the previous and the carefully sequenced instructions in the PowerPoint slides. In the post-lesson discussion, both teachers felt they needed more experience to teach with Scratch, and were in a disadvantaged position, since the children had studied ScratchMaths for a year. I made the decision that further collection of data would be abandoned, because of the teacher’s frustration and the supportive role that I was playing within the classroom. The teachers wanted to attend PD when offered again and would aim to teach the Year 5 curriculum the following year.

**Group B**

Group B, in contrast to Group A, had attended the PD and had either taught ScratchMaths in year 5 or worked closely with teachers that did. Matt, Christine and Nancy all taught activities using the *Fireworks* microworld.

**Nancy**

Nancy had a strong scientific background. She had science A Levels and an undergraduate degree in Biology. Nancy first encountered the idea of variable at secondary school. She explained that there are two types of variables: dependent and independent and provided a definition of each. Nancy reflected on when she had used variables in her ScratchMaths lessons:

> When you are inputting data and saying how many you are going to do, an independent variable how big your block is going to be in your tower, and your dependent variable is how tall your tower is going to be at the end of it. [INB3(Int)]

Nancy was asked to consider how her understanding of variable in Scratch bridged to variable in mathematics. She described a situation with her partner teacher who was not teaching ScratchMaths:

> I was quite good at telling her what we were doing, because it was interesting, we are doing this in SM at the moment, or come and have a look at what we’ve done this afternoon! She would get an idea of what we were doing, but she’s not able to do more with variable that we were doing, she wouldn’t be in a position to pull out those links as she didn’t have the intimate knowledge of it. [INB3(Int)]
Nancy was asked to explain what she meant by pulling out the links. She provided an example of how she typically used function machines, e.g., a number as an input goes into the machine, mathematical operations process the number and a number, an output comes out the other side. However, during the lesson she linked the function machine to using ask and answer and variables as in the Polygon Fireworks investigation:

I think it was the algebra part when we talked about it. I related it to some of the stuff we’d done, we talked about function machines. That’s how I introduce algebra if you put a number in and I get a number out. So, give me a number and this is how big it is, so imagine a tower, give me a number and this is what I give you. [INB3(Int)]

Nancy’s example described the relationship of the number of number of floors variable and the height of towers in the polygon fireworks investigation by linking the function machine representation to the representation of the Scratch language. E.g., the variable is the user input (of the function machine) which is then used to control the height using the repeat block. Nancy’s knowledge of variable in Scratch and in mathematics is undoubtedly well connected. She demonstrated the connections in the task-based interview using the variable task (See section 3.4.2.2) conducted three months after she had finished teaching the polygon Fireworks investigation. Table 8.1 shows what Nancy said as she was asked to ‘think aloud’ as she worked on the task. The right-hand column suggests a description of the aspect of Nancy’s MPTK demonstrated at each step.
Table 8.1 Sequence of Nancy thinking out loud as she worked on the interview based Scratch task

<table>
<thead>
<tr>
<th>Thinking out loud/ Scratch action</th>
<th>Scratch script</th>
<th>Descriptive analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>[She drags out a turn 15 block and a move block. She attaches the turn block under the move.]</td>
<td><img src="image" alt="scratch_code" /></td>
<td>The step indicates Nancy has formed a connection between the beetle’s turn through the exterior angle as it moves around the perimeter, and 360 degrees for a full turn. She used this mathematical fact to calculate the angle of turn for an octagon. She has recognized that the regular octagon has eight equal sides and calculated each from the perimeter.</td>
</tr>
<tr>
<td><strong>NB:</strong> Octagon, perimeter is 600. So, half of that is 300 … so half of that is 150… so each side is 75. [She edits the move to 75.]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NB:</strong> So, I’m going to move 75, and because it’s an octagon it’s going to be divide by 8. [She drags out a divide operator and edits it be 360 /]</td>
<td><img src="image" alt="scratch_code" /></td>
<td></td>
</tr>
<tr>
<td><strong>NB:</strong> So, I will need some questions and some variables. [She goes to sensing and drags out an ask block. She types in How many sides? She drags out a second ask block and types in What side length?]</td>
<td><img src="image" alt="scratch_code" /></td>
<td>Nancy confidently performed these steps and knew where to find the required blocks in the Scratch palette. She validated the variables had already been defined (due to the state of the Scratch project) and created the associated questions. She set the variables to the answer block, a key step in connecting user defined variables, to the user input, using the answer block. She has demonstrated that the answer block can be used multiple times to store the answers to different questions.</td>
</tr>
<tr>
<td><strong>NB:</strong> Set number of sides to answer and then set side length to answer. [She drags out two set blocks, changes each to a different variable, drags out an answer and inserts it into each set block.]</td>
<td><img src="image" alt="scratch_code" /></td>
<td></td>
</tr>
<tr>
<td>[She orders the ask and set blocks, so that the set statement immediately follows the ask block]</td>
<td>The step indicates that Nancy understands the relationship between ask and answer and must be stored in a variable – i.e., answer holds the input to the last question asked in the script.</td>
<td></td>
</tr>
<tr>
<td>[She drags out the variable side length and inserts into the move steps.] [She drags the variable number of sides into the operator block, and drags this into the turn block.]</td>
<td>Nancy performed these steps confidently. Her actions suggest that she can represent the general structure of a regular polygon using a move and turn, and can operate on a variable. She also demonstrated an understanding of using a variable to control movement. Finally, this step provides evidence to support the claim that Nancy is able to generalise the external angle turned by the beetle for any regular polygon.</td>
<td></td>
</tr>
</tbody>
</table>
Again, Nancy confidently performed each step of her programming without error.

She demonstrated that a variable can be used in multiple places within a script as shown by the `repeat` block.

---

[She drags out a `repeat` block and wraps it around the move and turn.]

She drags out the variable `number of sides` and puts it inside the `repeat`.

She connects the blocks together in sequence.]

**NB:** And then we go to events.

[She drags out a `when space is pressed` and attaches it to the top of the script and runs it].

[She runs the script.]

**NB:** 8 sides and 75 for side length.

---

**NB:** I can’t remember how to do this.

[She drags out a `say` block and an `addition` operator block.]

Nancy paused at this point unsure what to do, the confidence gone which she had shown through the previous steps. However, thinking on her feet, she says, “The perimeter is, the number of sides multiplied by” and
**NB:** The perimeter is, the number of sides multiplied by…

[She drags out a multiplication operator block and drags the side length and number of sides variables into the operator block.]

immediately drags out a multiplication operator block and successfully performs each step without pausing.

[She drags out a join block and changes the text to say “The Perimeter is”.]

[She drags the operator block into the join block].

[She drags the join block into the say block and runs the script].
In summary, Nancy’s response to the task demonstrated her confidence in using Scratch to represent the structure of a general polygon in Scratch using two variables and can use and operate on variables in different script contexts:

- Define and set variables,
- Use a variable to control the beetle’s movement,
- Operate on a variable in a calculation to control the turn the beetle’s turn,
- Use a variable to control the number of times the beetle repeats drawing each side,
- Use two variables in a calculation operating on one another.

The task outcome suggested that Nancy had made connections between the Scratch representation of a regular polygon and the geometrical figure in mathematics, as well as connections between variables in Scratch and in mathematics. As a summary to the task, Nancy was asked to explain how she would assess a child’s work mathematically if their final script matched, or was similar to hers:

To be honest it’s algebra, it would be nice to pull out the algebra and show that as an algorithm. I would express it algebraically, I would put in SL for side length, and we could look it as a formula really, and say that to find the perimeter, because that’s how I’m thinking, I need this length and that length multiplied together, so I would probably do ‘My perimeter is; this multiplied by this, and pull out a little of the algebra really. [INB4(Int)]

**Matt**

Matt did not make the same progress as Nancy in the variable task. His approach [BMS3(Int)] was less confident. He joined some blocks together, provided little explanation of this thinking and quickly broke apart his work. For example, he started the task and dragged out an ask and answer block and set the “How many sides?” question. He recalled that the angle of turn was 360 / angle and created the script shown in Figure 8.5

Figure 8.5 Matt’s initial script to draw a regular polygon which uses an ask block and an expression using answer to control the turn
Without running the script, he broke the script apart, and said “Ahh, the first question should be how many steps”. Matt’s response suggested that he noticed that the script for a general polygon was incomplete as it lacked any movement. He responded to this observation by dragging out a move block and replacing the number of steps with answer. He dragged out a repeat block and wrapped it around part of the script:

![Figure 8.6 Matt’s developing polygon script which asks a second question question for how many sides inside the repeat](image)

Again, without running the script, he broke the script apart and said out aloud, “If I was doing it normally, it would be move, turn, repeat how many times I wanted”. Matt’s response suggested that he can represent the general structure of a regular polygon in Scratch, i.e., move, turn, repeat. He rebuilds his script from Scratch to demonstrate this.

![Figure 8.7 Matt’s script to represent the general structure of a regular polygon, a repeat, a move and a turn](image)

However, the script still contains two bugs:

- Repeat 10 is a fixed number;
• **Answer** is used in the **turn** block and in the **move** block.

The bugs as represented in Scratch suggest that he had not considered variable to express the relationship between number of sides and turn, a key idea in expressing with variables, since **repeat** was a fixed number. Using the **answer** block twice may indicate that he holds the view that the block can hold the values to the different questions concurrently. E.g., answer inside the turn block is the answer to the “How many sides question” and answer inside the move block is the answer to the “How many steps” question. This was a common outcome amongst the children as they engaged with this activity during pilot testing of the ScratchMaths materials, and of researching of novice programmers understanding of variable and was therefore built into the teaching sequencing of introducing variable in Scratch (see section 4.2.2). Matt had taught this sequence which suggested that his knowledge of how variables can be used in programming is weakly formed. He is confronted with the limitation when he tried to move the ask block to be before the turn block as shown in Figure 8.8 and the dialogue:

![Figure 8.8 Matt exploring where to place the ask block as he confronts the limitation of not being able to store two answers simultaneously](image)

**MS:** That wouldn’t work as it would ask how many sides each time.

**PS:** Would variables help here?

**MS:** [Ignores the prompt] Move that number of steps, then turn the answer to that. The trouble is, fitting that in, it would ask the question every single time. I can see the problem, I’m just trying to think of my way through the solution.

Matt’s response to the problem was to break the script apart, and to then **explain** each block step by step as he rebuilt it. The use of **explain** is a key 5E pedagogical strategy, also observed in Matt’s lesson [BMS1(Obsv)]. However, despite this strategy he found himself stuck and faced with the same problem as previously, e.g., his script could not process two answers.
simultaneously, which suggests that does not know that user defined variables can overcome the limitation. To resolve the problem and complete the task, Matt chose to ignore the suggestion of using variables and instead fixed the side length with a static value. His final script is shown in Figure 8.9 below.

![Figure 8.9](image)

In summary, Matt’s response to the task demonstrated that he could use Scratch to represent the structure of a general polygon in Scratch using one variable, the *answer* block. He can use and operate on the *answer* block in different script contexts:

- Operate on a variable in a calculation to control the beetle’s turn
- Operate on a variable in an expression to calculate the perimeter,
- Use a variable to control the number of times the beetle repeats drawing each side,

The task outcome suggested that Matt has made a connection between the Scratch representation of a regular polygon and the geometrical figure in mathematics. He has demonstrated a link between variables in Scratch and in mathematics but has not developed an understanding of how to use user defined variable to overcome the limitation of the answer block to use and operate on two variables simultaneously. As a summary to the task, Matt was asked to explain how he would assess the task if a child had created a similar script. In this extract he demonstrated the importance of the children to be able to *explain* their work, rather than just accepting a successful outcome:
If they could produce that, at least secure, if not secure plus, but I’d need to talk to them. I’d ask them how they built it. Checking that they had not stumbled across or copied it, making sure that they could explain the divisions, and what they had chosen to divide by, and why they had chosen to multiply this and that. [BMS3(Int)]

8.2.3 Window on angle

Group A

Betty, Trevor, Elizabeth did not engage explicitly with the key concept of exterior angle that the beetle turned through in either of the observed lessons [HBK1(Obsv), GEA1(Obsv), GTT1(Obsv)] using the Polygon fireworks investigation. In this episode of Betty’s lesson [HBK1(Obsv)], she discussed how to make a triangle, after the children had successfully created a square. She discussed with the children that they needed to change the repeat to a 3 to represent the number of sides and asked the children to consider the angle of turn:

**BK:** Who knows, how many degrees are the angles of a triangle?
**Child:** 120!
**Child:** 180!
**BK:** [Ignoring both responses] How many degrees are the inside of a triangle? Matthew?
**BK:** An equilateral triangle?
**Child:** 180.
**BK:** 180 all together so how much would each angle be?
**Child:** 60.
**BK:** 60!

The dialogue suggested that Betty has not considered the angle of turn that the beetle makes and is relying on procedural mathematical knowledge of triangles, e.g., the sum of interior angles in a triangle is 180 degrees, a similar finding as shown in Sally’s case. Betty did not use the computer during this discussion which would have confirmed that there was a problem with her mathematical reasoning. Later in the lesson, Betty used the fact that the angle of turn for an octagon can be found by dividing three hundred and sixty by eight, however did does generalise or exemplify for the triangle. For Betty, it appears that this mathematical fact is isolated, unconnected to either the beetle’s turn or the specific case of the triangle.
Elizabeth demonstrated a similar reliance on procedural knowledge as Betty in her post-lesson interview. She was asked to explain how she would assess a child’s outcome from the lesson as shown in Figure 8.10 below. An extract of her explanation is show in this dialogue [GEA2(Int)]:

![Figure 8.10 Child's ScratchMaths polygon fireworks and towers.](image)

**EA**: He has done some lovely stars in the sky, fireworks, and rectangular shaped buildings.

**PS**: How might he draw the star? Where might it start?

**EA**: From the point on the left hand pointing downwards, then go up and go down to the right-hand side, then go up to the left, then across to the right and the back down.

**PS**: If we start at the bottom, and we can go up, that’s maybe move 30, let’s say, what do you think it has turned?

**EA**: … turn to go down?

**PS**: And what is that turn?

**EA**: 45 degrees? then go down.

**PS**: OK, and what would it need to turn next?

**EA**: I’ll keep it as 45 degrees and the same amount of steps across.

Elizabeth had indicated an internal angle, as represented by the tip of the star. Since she did not complete any of the professional development or teach ScratchMaths in year 5, it is unlikely that she is thinking like a beetle and imagining herself turning through the exterior angle. However, when she was asked to think what angle would be required to make an octagon and she responded that it would be “360 divided by 8, 40 degrees”. She is informed of her calculation error, that the angle of turn would be 45 degrees, i.e., the same as the angle that she has given for the star. Despite the error, Elizabeth was able to recall some procedural
knowledge about the angle that is required to draw an octagon, possibly from her taught lesson. However, Elizabeth does not notice that she gave the same angle of 45 degrees to both the star and the octagon questions, suggesting a very weak understanding of the exterior angle made as the beetle turns.

**Group B**

Nancy’s and Matt’s strength of geometric understanding, in particular their understanding of the total turn of the beetle being three hundred and sixty degrees, was demonstrated in their response to the variable task explained in section 8.2.2 above. Nancy and Christine’s understanding of the exterior angle the beetle turned through is exemplified in their use of playing beetle and chalking out the regular polygons which is discussed in section 8.3.2.2 below. Although Christine did not complete the variable task, in her post-lesson interview she was asked to explain why she had used $\text{turn } 360 / \text{number of sides} \text{ degrees}$ in the definition of a regular polygon. She explained “Because a full turn is 360, shapes are always a full turn, it just depends how many turns it has in it depending on the shapes side number” [ICC1(Int)]. Her response suggested a strong indication of a connection formed between the Scratch representation of turn for a regular polygon and the mathematical idea of exterior angle for a regular polygon.

### 8.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

[ScratchMaths] has really influenced my thinking about place value, ‘cos I think it was always one of those things, where I knew it was the case and explained it, but actually now with more of my Scratch head on, this happens and then this makes this change and that kind of thing!

I explained it to the Year 2's, using that kind of language… I showed them the model so that they could actually see another way of physically changing the numbers. [Christine, CKW(PVT-Q10)]

In this section, I discuss two themes which emerged from the analysis of the three major cases. Within each theme, I will illustrate how the data of some of the teachers from Group A and
Group B characterise the theme and explain the relationships between the mathematical idea and the programmatic representation (see Table 4.5 on page 122 for classification of each theme).

### 8.3.1 Thinking with your maths head on

Katherine had taught using the *Place value* microworld with the children in the lessons prior to the observation. At the start of the observed lesson [CKW1(Obsv)] Katherine used an activity she had designed to engage the children with one of the key features of the model: *when* to broadcast a message. She displayed the code of the ones sprite but had modified the code to contain a bug, e.g., `if costume # = 9` rather than the correct `if costume # = 10` as shown in Figure 8.11 below. Katherine focussed on asking the children to *explain* their reasoning as illustrated in this episode. She started the interaction by asking the children “We are going to look at a script to see if it’s going to count up, and if it doesn’t, I want you to tell me why.”. The children discussed the problem with their partners, and Katherine asked the children to vote who thought it would “count normally”. A few hands were raised, and Katherine picked a child to explain:

![Figure 8.11 Four-digit place value model used by Katherine edited to contain a bug, e.g., if costume # = 9](image)

**KW**: Alex thinks it will. Can you explain why?
**Alex**: Because it’s got next costume.
**KW**: When I click the ones sprite, what will happen?
**Alex**: It will go to the next costume.
How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

KW: Which will be?
Alex: Number 1.
KW: When I get to a 9, what will happen there?
Alex: It’s going to broadcast a message to the tens to tell it to change.
KW: OK, do we agree with Alex?
Children: No.
KW: Lots of sitting on the fence, Yassin?
Yassin: We don’t know about the tens sprite if it will catch that message.

At this point of the interaction, Katherine has encouraged the child to develop their reasoning beyond the simplistic, “it will change to the next costume,”. The children correctly identified that in the Scratch program when the digit sprite changes to a nine it will broadcast a message. However, they have not yet made the connection that this would be a bug for a base ten place value model, a key idea for this model. The children had correctly explained the broadcast and when I receive function; one child corrected the other child by explaining that the message is not broadcast to the tens sprite, but that the tens sprite needs to be programmed to catch or receive the message.

In the next part of the episode, Katherine continued to encourage the children to explain their thinking:

KW: Before I show you, a couple more opinions, Shashi?
KW: Explain your thinking. I think that something has just happened in your brain, and I want to know what that thing was!
Shashi: I was thinking that when you put costume 9, it would change to 10, but…
KW: Why do think that. Use the scripts to help you to explain?
Shashi: It doesn’t say… [She trails off.]
KW: Before I show you, go on Daniel you’re dying to tell me.
Daniel: It says if a costume is 9 then it will change, but in real life it changes when you go past 10.
KW: So, are you saying this is where my problem is? Let’s test it... what is going to happen?

In the dialogue, a child had made a connection between the place value model in Scratch and the place value model in mathematics, or in “real life” to use the child’s language. Katherine does not confirm the explanation, but instead suggests the children watch as she ran the
program. She clicked the *ones* sprite so that it counted up and the display changed from 0001 to 0009. When she clicked once more from 0009, the display changed to 0019. She asked the children to *explain* what they had seen.

**Efisa**: Because when you change to 9, you broadcast it.

**KW**: Do we want to broadcast a message when it is a 9?

**Children**: No.

**KW**: That’s the problem isn’t it? Remember when we were playing the game? Noah was nudging, when did Noah nudge? Yassin?

**Yassin**: When it goes to 0.

**KW**: What does that need to say?

**KW**: 0 or 10, why?

**Children**: *in unison* 10.

*A child tries to explain that you need to know the costume number for the 0 digit, but the exact dialogue is inaudible*.

**KW**: Who can remember, what I use to help me with this predicament?

**Ryan**: Go on looks, and press costume #.

**KW**: And that helps, and it displays the costume number.

*A child tries to explain that you need to know the costume number for the 0 digit, but the exact dialogue is inaudible*.

**She checks the tick box next to the costume # in the palette, clicks the 9 once more. The display changes from 0019 to 0010. **ones**: costume # 10 is displayed for the selected ones sprite as shown in the top left corner of Figure 8.12 below*.

![Figure 8.12 Scratch stage showing costume # reporter block reporting a value of 10 for the digit costume (0) worn by the ones sprite.](image)

The next episode of the lesson further exemplified Katherine’s strategy to encouraging the children to *explain* the connections between time, place value and the Scratch representation.
In the next activity, SM Activity 4.2.1, Katherine asked the children to open the new project, to *explore* the scripts inside the project, and to think about what they are going to build, as shown in Figure 8.13 below. She explained that the investigation was very similar to the previous investigation, building the place value model, but with some key differences. She asked the children to follow the instructions on the PowerPoint slide.

However, after the children had been working for a few minutes, she noticed that many had named their sprites, *ones, tens* and *hundreds* as they had done in the previous place value model, and not used the place names to represent time for a stopwatch. She stopped the children and asked the children to *explain* their reasoning:

**KW**: I’m interested in what some of you have named them...

**KW**: Let’s just take our Scratch heads off for a minute and put our Maths heads on. If this was a clock, what would this number represent?

**KW**: If we are thinking of the place value of time, what’s this Andrew?

**Andrew**: Ones.

**KW**: That’s the one second, what’s next, Yassin?

**Yassin**: Minutes.
**Teacher episodes**

KW: That’s not minutes yet is it, Amy?
Amy: Ten seconds.
KW: And then, what do we go to Yassin?
Yassin: Minutes.
KW: How many minutes though? That’s the one minute and then we’ve got the?
Child: 10 minutes.
KW: We are not in the hours yet are we? Make sure you have all labelled it correctly.

In this dialogue, two statements stand out, Katherine asked the children to “Take off our Scratch heads and put our Maths heads on” and “If we are thinking of the place value of time”. Both pedagogic strategies arise from her teaching with the place value microworld and indicate how Katherine bridged effectively between programming and mathematics. She used the strategy to link the Scratch programming structure representation of place in the base ten model with that of time in the stopwatch model. She linked the mathematical ideas of place value and time, and then made a link between the Scratch representations of place value and time and the mathematical idea.

### 8.3.2 Thinking like a sprite

In this section I present the theme *Thinking like a sprite* as illustrated in the contexts of variable and angle which emerged from the analysis of the three cases. Within each context, I illustrate how the data characterises the theme (see Table 4.5 on page 122).

#### 8.3.2.1 Thinking like a Beetle for variable

In this teaching episode from [ICC1(Obsv)] Christine introduced the concept of variable using an activity she had developed with her partner teacher Nancy. The activity was an adaptation of the ScratchMaths materials, Activity 5.1.3 (see section 4.2.2) using the *Fireworks* microworld. Christine and Nancy’s adaptation used the key idea of asking questions one after another to explore the impact on the answer stored by the program, before they addressed the specific problem as shown in the PowerPoint slide. They chose to model the activity using Post-it notes and the idea that the children would think like the beetle. Christine started the activity by asking the children to suggest questions that the beetle could ask to draw the Polygon firework display. The children offered:
8.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

- How many sides?
- What side length?
- What pen size?
- How many polygons?

She built this in Scratch as shown in Figure 8.14 below.

![Figure 8.14 Script which asks four questions to demonstrate how each successive ask block overwrites the previous value of answer stored](image)

As each child provided a question, they were asked to stand at the front of the class in a line next to each other and to hold a Post-it note. Christine explained that the Post-it note would hold the answer to their question when the question was asked in the script. She then asked each child their question and they told to write their answer on to the Post-it note as shown in Figure 8.15 below.
To start the demonstration, she asked, “So now if my script has just begun, what will it ask me, Adam?” (Adam is the first child in the line, holding up a 6).

**Adam**: How many sides?

**CC**: 6 please.

[She writes 6 on a post it and stands in front of Adam holding a 6.]

**CC**: I move along the script, what’s next?

**Brianna**: How long the length is?

[As Brianna says this, Christine writes 10, the number Brianna was holding on to a new post it and scrunches up the previous post it.]

**CC**: What’s happened?

**Child**: It’s a waste of paper!

[She ignores the comment and continues moving down the line of children.]

**CC**: What pen size?

[She screws up the post it.]

**Child**: That’s disgraceful!

[She ignores the comment.]

**CC**: I need an answer, I need to store an answer!

**Chelsea**: 7.

[She writes 7 on a new post it.]

**CC**: So, at the moment, how many sides is my shape going to be? If my answer is this?

The final question Christine asked in this dialogue is key to examine the children’s understanding of how Scratch uses the **ask** and **answer** blocks, i.e., when a new question is asked, the **answer** block stores the value and overrides any existing value. Christine’s selection of the question suggests that she had understood how variables were set using the **ask** and **answer** blocks as a foundation to introduce user defined variables.
8.3 How mathematical ideas mediate and are mediated by engagement with the ScratchMaths curriculum

Returning to the classroom dialogue, Christine used the question “So at the moment, how many sides is my shape going to be? If my answer is this?” to encourage the children to think like a beetle i.e., to imagine that they are the Beetle asking and processing the questions step by step. For example, the first child has a 6, for number of sides, which is overwritten by the number 10 (the child who asks side length), and then overwritten by the number 7, (the child who asks the pen size). At this point of the children’s reasoning process there are three possible answers for the value of the side length, 6, 10 or 7. Christine responded to the children’s responses:

**Child:** 6, no 7.

**CC:** And what’s my side length going to be?

**Child:** 10!

**CC:** Is it?

**Child:** 7!

**CC:** I’m going to have a seven-sided shape, it’s going to have length of side 7, it’s going to be 7 thickness, but hold on a minute. I’ve got a new answer!

*She screws up the previous post it, as she moves to the last child in the line.*

**CC:** How many shapes do I want? I want 20 shapes. So, can somebody tell me what’s just happened?

**Child:** Once you answered a question, and answered another one, the answer from before goes!

**CC:** Yes, because I could only store one answer at a time!

**CC:** So, if you write lots of questions in your script and put answer and answer, it will always use the number you’ve put in for the last question, the most recent question.

**CC:** It doesn’t matter that you wanted a six-sided shape Adam, because I can only store 20-sided shape in mine at the moment. It’s also going to be 20 side length, and also going to be 20 thickness and it’s going to be 20 of them. Because I can only store one answer at the same time.

In the final part of the dialogue, she moved to the last child holding a Post-it note with a value of 20 and reiterated that the answer block contained a value of 20, the value of the last question asked. Following the use of the Post-it note representation she engaged the children with the how they could create user defined variables so that multiple values could be stored simultaneously.
8.3.2.2 Thinking like a Beetle for angle

Both Nancy and Christine used the playing the beetle approach to demonstrate the angle turned by the beetle. Nancy described a lesson where she drew diagrams on to the classroom floor, prior to engaging with the Scratch language.

[I wanted them to engage with] Straight lines, external angles and internal angles. So, I would draw the shape on the floor using masking tape.

For example a hexagon, I chalked it out, sort of extended the shape, the sides if you will [to show the exterior angle] … I then drove a highlighter along it, I wanted a toy car, but I didn’t have any, so I drove a highlighter along it. As I turned the highlighter, I drew the angle in at the same time. I ended up with six external angles and we thought about the six turns that I had made. [INB3(Int)]

She explained how she discussed the specific example of a pentagon to reinforce the exterior angle turned:

In in Scratch, because of the way that it turns it’s obviously not turning an internal angle… For example, if you are looking at the pentagon you can see that you’ve got 5 obtuse angles, so there’s no way it’s going to be 72. [INB3(Int)]

Nancy referred to this episode in her observed lesson [INB1(Obsv)]. She used the approach to support the children to explain how they would build the skyscrapers towers of squares. She asked a child to direct another child playing beetle at the front of the classroom. The child instructed the beetle child to move to a random location on the floor, to put the pen down, to draw a square, and then to ‘repeat it’. Nancy interrupted the children:

**NB:** What’s this bit, what’s she doing then? [Referring to the child moving forward after drawn one square].

**NB:** So she’s drawing a square and then she’s doing something else? What’s she doing after that?

**Child:** She’s moving 30 steps.

**NB:** Why do you think, you built this Kaylee, so why do you think she’s having to move 30 steps?

**Child:** So you can do another square on top of it.

[**Nancy shows the script on the computer.**]

**NB:** OK. I’m just going to show you. I’m going to take the little bug, and click square, pop pen down first, so there’s my square. And we are here, at the bottom of the square, so what’s going to happen when we draw another square?

**Child:** Go on top of it.
**NB:** So what we need to do is once we’ve built our square, we need to move up a little bit. We need to shift up a little bit. Now you’ve said 30, so what if now, I wanted to build a tower with different sizes of squares.

The dialogue suggested Nancy has made connections between the representations of angle and movement acted by the beetle on the floor and angle and movement of the beetle inside the microworld. Her continued use of the ‘Play Beetle’ strategy, beyond what was written into the task, suggests a developed confidence with using the representation to connect the mathematics with the programming and vice versa.

### 8.4 Teacher pedagogic practices

In this section, I identify and exemplify the kinds of whole-class activity types utilised by the teachers from Group A (little previous ScratchMaths experience or PD) and Group B (taught ScratchMaths in Year 5 or worked closely with Year 5 teachers). I then go on to analyse the quality of the orchestrations and the adherence to the ScratchMaths curriculum using the adapted TRU framework for teaching mathematics through programming as discussed in section 3.6.4.
8.4.1 Whole class orchestrations

Table 8.2 summarises the total time the teachers spent on each whole-class orchestration within an observed lesson of approximately an hour. The remainder of each lesson time was allocated to children working at the computer either independently or in pairs.

<table>
<thead>
<tr>
<th>Group</th>
<th>Teacher</th>
<th>Sherpa-at-work</th>
<th>Link-Multimodal-represent..n</th>
<th>Discuss-the-screen</th>
<th>Explain-the-screen</th>
<th>Board-instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Betty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Anthony</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Priya</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elizabeth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trevor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Laura</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Katherine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Matt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nancy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Christine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2 Cumulative time for each whole-class orchestration in one lesson.

<table>
<thead>
<tr>
<th>Time Duration</th>
<th>Group A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 mins</td>
<td></td>
</tr>
<tr>
<td>&lt;5 mins</td>
<td></td>
</tr>
<tr>
<td>5≤ t &lt;10 mins</td>
<td></td>
</tr>
<tr>
<td>10≤ t &lt; 15 mins</td>
<td></td>
</tr>
</tbody>
</table>

**Group A**

A first observation of the table is the absence of technology orchestrations in Anthony’s and Trevor’s observed lessons. Neither teacher used live coding to teach Scratch despite the availability of the technology. Trevor’s teaching approach [GTT1(Obsv)] was to display the ScratchMaths PowerPoint slides on the IWB and to tell the children what to do. Although he engaged in whole-class discussion with the children and frequently supported the children in a one-to-one setting at their computers, he did not use technology to teach the whole-class. Trevor’s decision to not use technology is perhaps not surprising given the challenging resourcing situation at the school. His lack of ScratchMaths PD was due being assigned to
teach ScratchMaths after a member of staff had gone on long term sickness. He was also working as a teacher which is beyond his training as a Higher-Level Teaching Assistant. His case is indicative of the challenging circumstances of some schools and was perhaps exacerbated since the school wanted to support the research study they had signed up to.

In Anthony’s lesson FAO1(Obsv)] he did not use the ScratchMaths PowerPoint slides. Instead, he showed the children a fifteen-minute video explanation of how to build the place value model which I had been created for the SM intervention as a PD webinar. In the recorded video I explained step by step how to build the place value and the decisions made at each programming step. Whilst the video played the children were expected to take notes. Anthony stopped the video frequently to tell the children to record a specific step such as “Duplicate the one sprite and rename it to tens” or “Broadcast an add 10” message. Following the video, the children moved to the computer room to create their own place value model. The children worked on the computer for the rest of the lesson. Some children were provided with one to one support by Anthony; there was no further whole-class teaching episodes. Anthony explained after the lesson that he had tried to teach the place value model in his previous lesson but was unsuccessful due to his limited Scratch knowledge. He therefore decided to use the video in place of him using the computer:

I thought the video would be a lot more informative, than me doing it. Last week [he laughs], they did have some issues. So, when I saw that [video], it seemed a lot clearer. [FAO1(Int)]

The lesson outcome was perhaps unsurprising, given the difficulties Anthony had with completing tasks within the PD sessions. I observed that he was unconfident and relied on his partner teacher, Priya who although similarly inexperienced, was more confident to explore using Scratch and could support him. It is unclear whether Anthony continued to teach ScratchMaths beyond the observed lesson and whether the lesson was staged to comply with the initial research agreement.

However, at the same school, Priya (Anthony’s partner Year 6 teacher, observed immediately after Anthony), did use the technology based whole-class orchestrations Sherpa-at-work and Discuss-the-screen. In the lesson [FRP1(Obsv)] she created the digit cards to play the Nudge, Nudge, Flip, Flip game with the children and used live coding to start the children off to build the Scratch place value model. Her use of the digit cards is an example of the Link-multi-modal-
representation. Although she had found the PD challenging, she explained that she had tried the activities again outside of the PD. Priya also explained that despite teaching the same year group as Anthony, she was unable to support Anthony since they were not free at the same time and so had little opportunities to plan together.

A second observation of Table 8.2 is the similarity of the whole-class orchestration profile for Elizabeth and Betty. Both teachers used technology for very short episodes as shown by their use of the Discuss-the-screen and Explain-the-screen (Betty only) whole-class orchestration types. They additionally spent about 10-15 mins of the lesson time using Board-instruction without the use of technology. Both teachers had not attended the PD and were only aware of a limited amount of the support materials available. For example, Betty was aware of both the SM PowerPoint materials and the SM Teacher guide but did not use the starter projects. The consequence for Betty was that she used Scratch in its default state, i.e., an empty new project which only contains the Scratch sprite. The children found it difficult to find out how to clear the screen after drawing and to return the sprite to the centre of the screen since there was no pre-built set up script to perform these functions, as is found in the ScratchMaths starter projects. The default cat sprite’s perspective is viewed from the side which was particularly problematic when combined with the ScratchMaths PowerPoint slides which had all been designed to use a beetle sprite viewed from above.

In summary, Group A, except for Priya, made limited or no use of technology during whole-class teaching episodes in the lesson. The teachers chose to tell the children what to do or held discussions about what the children did during their independent activity without demonstrating or showing children’s work. However, Priya’s use of technology appears to be an outlier within this group, despite having a similar teaching experience background and experience of ScratchMaths. She was clearly better prepared than the other teachers from this group since Priya was the only teacher that had created additional support materials for the children. For example, she printed a sheet which contained code for the 10 seconds sprite which could be copied to scaffold the children’s programming as they built a working stopwatch. Priya had also created a mini lesson-plan and wrote notes on a Post-it Note. All other teachers in this group used the PowerPoint slides as their guidance, with no additional notes. I am assuming that all teachers had planned their lessons. The post-lesson interviews [FAO1(Int),
HBK1(Int), GEATT1(Int)]) provided evidence that the teachers had tried the activities for themselves, but did not create any form of lesson plan, which seemed to suggest some preparation. However, in Anthony’s lesson, his choice to use of the video instead of live coding indicated that he was currently unable to provide any performance on his own confidently.

**Group B**

An initial observation for the Group B teachers is the similarity of their use of the technology whole-class orchestrations. Each teacher spent at least ten minutes of their teaching time using Scratch live in front of the children and used the Explain-The-Screen and Discuss-The-Screen orchestrations. The teachers coded live, changed values within the blocks, ran scripts to test out suggestions in either the Place value or Fireworks microworlds. Christine used a greater amount of time using the Discuss-The-Screen orchestration because she taught a smaller group of children and tended to work with them simultaneously. When she worked in this way, she coded live, the children would work independently, they would stop to discuss what they had done, and move on together. The consequence of this whole-class approach is that she spent a longer time in the lesson discussing the screen than the other teachers in Group B, before she would allow the children to work independently.

A second observation is the use of the Link-multi-modal-representation, an orchestration used by all the teachers except Katherine. Christine and Nancy used Post-it Notes to support the development of the concept of variable and is discussed in more detail in the following section. Nancy played beetle with the children herself and facilitating two children to instruct each other to play beetle. Matt used role-play to act out the effect of the order of block placement in a script. Laura used small digit cards to represent the functionality of a conversion game, first played in pairs before the children built the same functionality in Scratch. Despite the lack of the orchestration in Katherine’s lesson she referred to using the digit cards in a previous lesson:

> I don't make it obvious to them, when we first started with the flip cards, I didn't tell them that we were focusing on place value, we just played the game and looked at it without explaining. I related it back during the lesson when we worked on Scratch. [CKW1(int)]

In summary, the teachers of Group B use of technology orchestrations is not unexpected given their previous experience with either teaching the curriculum in the previous year or working closely with the teachers that did. What is more surprising is the teachers’ decision to use Link-
multi-modal-representation to supplement the forms of representation designed in the ScratchMaths materials. Envisage is a core E within the 5E pedagogical framework and exemplified within the materials as playing beetle or envisaging the output of the code before running. However, the use of roleplay in Matt’s lesson as shown in this episode was not a suggestion in the materials.

**MS:** Imogen, I want you to stand up. I’m going to give you some instruction. I want you Imogen to wait 1 seconds and then crouch down on the floor, from when I click my fingers.

[MS clicks his fingers and Imogen waits one second and then crouches down to the floor.]

**MS:** Great, now stand back up again

**MS:** Now, when I click my fingers, I want you to crouch down on the floor and then wait 1 second.

**MS:** What’s the difference? What’s the difference?

**Child:** The first one was quicker.

**Child:** Wait then go down and then she done it straight away.

**MS:** Good straight away, there we go. It is the difference between doing something straight away and waiting to do something. So, when I click my seconds sprite, am I going to tell it to wait a second and then change its costume or tell it to change its costume and then wait a second.

**MS:** If I’m counting backwards and I want to keep pace with all the other sprites, do I need it to wait, or do it and then wait. I need it change and then wait.

Similarly, Nancy and Christine’s use of Post-it Notes was developed as they engaged with the idea of variable in Scratch.

### 8.4.2 The TRU framework

Table 8.3 shows the scoring for each dimension for each teacher’s lesson using the adapted TRU rubric. The framework’s numerical scores represent a Basic (1), Proficient (2) and Distinguished (3) performance in each dimension (see Appendix A for the full scoring rubric). The results for each observed lesson are shown in Table 8.3 and represented as a chart shown in Figure 8.16 and Figure 8.17.
### Group A

<table>
<thead>
<tr>
<th>Teacher</th>
<th>Code</th>
<th>Mathematics through programming</th>
<th>Cognitive demand</th>
<th>Equitable access to content</th>
<th>Agency, Ownership and Identity</th>
<th>Formative assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betty</td>
<td>HBK1(Obsv)</td>
<td>1</td>
<td>1</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Anthony</td>
<td>FAO1(Obsv)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Priya</td>
<td>FPR1(Obsv)</td>
<td>2</td>
<td>2</td>
<td>2.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Elizabeth</td>
<td>GEA1(Obsv)</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Trevor</td>
<td>GTT1(Obsv)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Group B

<table>
<thead>
<tr>
<th>Teacher</th>
<th>Code</th>
<th>Mathematics through programming</th>
<th>Cognitive demand</th>
<th>Equitable access to content</th>
<th>Agency, Ownership and Identity</th>
<th>Formative assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laura</td>
<td>ALH1(Obsv)</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.5</td>
<td>1</td>
</tr>
<tr>
<td>Matt</td>
<td>BMS1(Obsv)</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Nancy</td>
<td>INB1(Obsv)</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Katherine</td>
<td>CKW1(Obsv)</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Christine</td>
<td>ICC1(Obsv)</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 8.3 Scores for each dimension of an observed microworld lessons using the adapted TRU framework for Maths through programming.
Dimension 3, Equitable access to content

Contrasting the scoring of Group A and Group B for dimension 3, Equitable access to content, highlights a key difference between the teachers. All teachers from group B, and Priya from group A had created classrooms that were well-managed. The teachers had established clear routines with their children for participation and structured the lesson into a series of teaching episodes which focused on one or two key ideas. Typically, after computer-based activity, the teacher would discuss the previous activity and ask the children to explain their progress. The teachers were careful to include quieter children and those children who had struggled in the task. Common problems were usually identified and addressed by the teacher in discussion with child in a whole-class setting. However, in Group A’s lessons, there was frequent calling out from the children, which although commented on by the teacher, tended to persist. The loudest and seemingly most confident pupils were selected to share their work and little attention was made to ensure even participation. During computer-based activity, many children were often off task and did not request help from the teacher. Many children in these lessons did not have an opportunity to have their work checked by the teacher, often resulting in a misunderstanding of the task. E.g., in Betty’s lesson, discussed in section 1.3.1 above, the children had not engaged with the answer block, in the ask and answer task, resulting in the children using literal text to display their name on the screen, and think they had completed the
task. In Anthony’s lesson the children worked for about 40 minutes with little input from the teacher.

From the scoring rubric:

**Group A:** Basic: *There is differential access to or participation in the mathematical content, and no apparent efforts to address this issue.*

**Group B + Priya:** Distinguished: *The teacher activity supports and to some degree achieves meaningful mathematical participation.*

**Dimension 1, Mathematics through programming**

Group A’s lessons (with the exception of Priya) are characterised by little or no use of the 5E’s pedagogic framework. Whilst each lesson did feature children working on the computer, their activity was largely unstructured. Elizabeth, Anthony and Trevor predominantly told the children what to do with little use of live coding to demonstrate or to test out suggestions from the children. Where there was some computer demonstration from Elizabeth, it was largely directive or lacked explanation e.g., she ran the final fireworks project to demonstrate what the children should be working towards but did not discuss in any depth how to get there. Betty used Scratch to demonstrate some ideas of ask and answer, but these appeared to be unfocused largely due to her lack of knowledge of and engagement with the ScratchMaths curriculum materials. Although Elizabeth, and Trevor taught the lesson in a large computer lab, with rows of computers facing the front, there was no attempt to encourage the children to talk to one another i.e. Exchange. Anthony’s lesson was an extreme example of telling what the children to do though his use of the video which instructed the children how to build the place value model. Elizabeth provided some limited opportunities for the children to explain an unplugged task, however she did not support the children to develop their reasoning or to make mathematical connections in their responses.

Group B + Priya’s lesson in contrast engaged with the 5E pedagogical framework and provided opportunities for the children to Explain their thinking and to Explore Scratch projects using the computer. All teachers except Katherine engaged with the multi-modal-representation orchestration to encourage the children to Envisage through the use of either the digit card
activity, playing beetle, the Post-it Note activity, or using roleplay to act out a script (Matt). However, Step, at the start of her lesson [CKW1(Obsv)] showed the script for a sprite in the place value model and asked the children to discuss with one another whether it would work or not; a task which exemplifies the Explain, Envisage and Exchange approaches. Across the observed lessons there was some evidence of the teachers making links to the mathematics, although in all cases with the except of Nancy these were fairly limited. Katherine began to bridge between the contexts of a stopwatch and the place value of time in her lesson, when she said “we are thinking of the place value of time” [CKW1(Obsv)] but the mathematical idea was not exemplified further. Nancy, on the other hand, frequently bridged to mathematics. For example, in an episode of the lesson [INB1(Obsv)], she explained how to build a skyscraper tower of squares. Nancy acted out how the beetle moved on the floor, to demonstrate how a variable in the script controlled the size of the square and how the same variable amount was needed to be moved to put one square on top of the other.

From the scoring rubric,

Group A: Basic: The teacher makes limited or no use of technology: simply tells the students what to do with no clear aim in mind. Student computer activity is unstructured, unfocused and with little or no use of the 5E’s.

Group B + Priya: Proficient: The teacher uses orchestrations and engages with the 5E pedagogical framework but connections with mathematics (bridgE) are limited. Students are given some opportunities to Explore, Explain, Envisage, and Exchange when working with the computer.

Dimension 2, Cognitive demand

Dimension 2 represents engagement with the ScratchMaths curriculum and the extent to which the teacher adapts or supplements the materials for their class of children. As can be seen from the chart show in Figure 8.16 above, Priya in Group A engaged with the materials consistently in her lesson [FPR1(Obsv)] and created a support printout sheet of the code for one sprite to enable her weaker children to start the task. Elizabeth used some of the materials in her observed lesson for the polygon fireworks tasks [GEA1(Obsv)], whereas Anthony, Betty and
Trevor made no use of, or were unaware of the teacher guidance, allowing the children for the most part to work on their own activity, which was weakly structured or constrained. For example, in Anthony’s lesson [FAO1(Obsv)] the children worked on creating a place value model but were provided little support or guidance other than the introductory video.

In Group B, all teachers used the ScratchMaths materials consistently within the lesson, in the intended sequence. Nancy and Christine created an additional activity and supplemented their introduction of variables with the Post-it Note task discussed in section 8.3.2.1 above.

From the scoring rubric:

Group A lies somewhere between Basic: Classroom activities are structured so that students mostly apply memories procedures and/or work routine exercises and Proficient: Classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to “scaffold away” the challenges, removing opportunities for productive struggle.

Group B lies somewhere between Proficient: classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to “scaffold away” the challenges, removing opportunities for productive struggle and Distinguished: The teacher’s hints or scaffolds support students in productive struggle in building understanding and engaging in mathematical practices.

Dimension 4, Agency, Ownership and Identity

The classrooms of the teachers from Group A were primarily places where the teacher initiates conversations. The teacher called on children to participate by name or responded to those who raised their hand first. The teachers had strict control over the classroom conversation. When students were asked to explain, their responses were short. The teacher was not always able to acknowledge the correctness of the child’s answer. For example, in Elizabeth’s lesson in her discussion of different scripts which each drew a square, but had different features such as colour or thickness due to the placement of the blocks, Elizabeth’s response was “That’s really interesting”. She did not challenge the child to explain their thinking behind their envisaged output. Priya’s lesson showed some evidence of asking children to explain their thinking,
however she accepted short responses from the children which were not sufficiently reasoned (or challenged), which suggests her developing confidence and knowledge with this task.

All the teachers from Group B provided significant opportunities for the children to explain their thinking. Children’s answers varied in length; however the teacher challenged the children to explain their thinking. For example, in this short episode from Matt’s lesson [BMS2(Obsv)]. Matt had asked the children to envisage the output of a short script:

**MS:** How do I know the script won’t make a triangle, or a rectangle?
**MS:** Quite a few hands, let me pick from the pot.
[Matt selects a lolly stick with a random child’s name on it from the pot.]
**MS:** Let me repeat the question, how do I know it will make a square?
**Child:** It’s repeat 4, so it will make 4 sides.
**MS:** So far, we know it will create 4 sides. What’s the last big clue?
**Pupil:** Because it’s got the 90 degrees.

From the scoring rubric:

**Group A:** Basic: *The teacher initiates conversations. Students’ speech turns are short (one sentence or less), and constrained by what the teachers says or does.*

**Group B:** Proficient: *Students have a chance to explain some of their thinking, but the teacher is the primary driver of conversations and arbiter of correctness. In class discussions, student ideas are not explored or built upon.*

**Dimension 5, Formative Assessment**

The use of formative assessment in Group A’s lesson closely followed the measure of the children’s agency and ownership as in the previous dimension. For all the observed lessons in this group, student reasoning was not actively surfaced or pursued. On the few observed instances in Elizabeth and Priya’s lessons the teacher action was limited to corrective feedback or encouragement. Priya, showed some indication that she was considering student thinking and common mistakes, but there was not sufficient evidence observed in the one lesson to confirm this observation.
Group B used formative assessment consistently in the observed lessons. Common mistakes were picked up during the teacher’s walk and referred to in the whole-class discussions. However, instruction rarely required the teacher to adapt their lesson to respond to children’s misunderstanding, i.e., there were no observed instances which required the teacher to “think on their feet” and respond to unexpected children’s responses. In Laura’s lesson the first unplugged task from the lesson was not marked or discussed as a class, however, there was some evidence of Laura supporting some children during her walkaround.

From the scoring rubric:

Group A’s use of formative assessment, *Basic*: *Student reasoning is not actively surfaced or pursued. Teacher actions are limited to corrective feedback or encouragement*. Note: Priya demonstrated that she was beginning to refer to student thinking and coming mistakes.

Group B’s use of formative assessment *Proficient*: *The teacher refers to student thinking, but specific student’ ideas are not built on (when potentially valuable).*

### 8.5 Summary

In this chapter, I have described and analysed teaching episodes and interview tasks for two groups of teachers. Group A teachers had attended some or none of the Y6 professional development and had not taught the Year 5 ScratchMaths curriculum. Group B teachers had attended the majority of the ScratchMaths Y5 and Y6 professional development and had taught the Year 5 ScratchMaths curriculum or worked closely with the teachers that had.

It is clear from the analysis of the data, that the teachers who had little professional development, or those that did not engage with the ScratchMaths teacher materials resulted in poor engagement with using Scratch *live* and taught unstructured and unfocused lessons. Anthony’s relatively weak knowledge of place value, combined with his lack of confidence and experience of teaching with technology, resulted in a lesson which had used a video demonstration followed by the children working mostly unsupported and unstructured for large part of the lesson. Elizabeth and Trevor who had wanted to try to teach the activities, despite lacking any formal PD, grew frustrated teaching an unfamiliar curriculum and teaching using
Scratch which they had little experience of. They were unable to support the children through the activities and their approach was mostly to tell the children what to do. However, despite similar challenges, Priya did make considerable progress. She had a similar amount of classroom experience as the group of teachers, had no prior experience of using technology and had attended the PD. The significant factor explaining her progress, is largely her engagement with the ScratchMaths materials and a commitment to careful preparation. This is also arguably what Betty had tried to do although she did not understand how the SM Teacher guide connected to the PowerPoint slides, and also missed the key idea of how the beetle moved.

There were some notable differences between the group’s mathematical knowledge of place value, variable and angle. Both groups of teachers could identify misconceptions in the children’s place value tasks. Anthony in group A provided the weakest weak answers to the items identifying the error but nor offering any explanation of the error or suggestion of what the child needed to know. Teachers from group B identified the error as well as providing a more nuanced explanation of the cause of the errors. With respect to variable, Betty from group A, taught a lesson which showed engagement with the Scratch Ask and Answer blocks but then introduced conditional statements, such as if then else, a complex computational concept, without attention to their purposed and meaning. The use of these blocks are not part of the SM curriculum and her decision to introduce them is an example of the dangers of a teacher trying out blocks which snap together easily and have a literal meaning, without careful attention to their purpose within computer programming. She had not attended any PD and was unaware of the SM Teacher guide. She was unaware of the 5E pedagogical framework which would have provided her with an opportunity to explore the blocks and to explain their purpose. Consequently, she created a trajectory for variable which she had not planned and moved further away from how the concept of variable concept had been carefully introduced, constrained and developed over the sequence of ScratchMaths activities as described in Chapter 4 section 4.2.2.

For Elizabeth and Trevor from group A who also taught the fireworks microworld there was no observable knowledge of their engagement with variable concepts within their lesson. Most of their whole-class activity discussion consisted of telling the children what to do to create
firework patterns, rather than explaining why they were doing it. They did not make any connections to mathematics. Discussing with the teachers after the lesson, confirmed that they felt they needed more professional development, and although they felt that the children had been successful as they had created some interesting patterns, they were not able to explain what the children had created. However, the group B teachers who had received PD and/or taught ScratchMaths in the previous year, demonstrated strong MPTK for variable concepts.

Table 8.4 below provides a summary of the findings and their location within each section of the chapter where they are exemplified.
Table 8.4 Table to show summary of findings and location within the chapter for the Stetches case study

<table>
<thead>
<tr>
<th>Content / Theme</th>
<th>Description – Group A</th>
<th>Section</th>
<th>Description – Group B</th>
</tr>
</thead>
</table>
| Place value     | • Identified children’s errors in place value tasks.  
• Variable depth of teachers’ explanation of children’s place value misconception. | 8.2.1 | • Identified misconceptions in children’s place value tasks.  
• Used Scratch model to identify children’s understanding (and possible misunderstanding) of written addition algorithms.  
• Explained programmatic structure of Scratch place value model using broadcast & receive.  
• Developed connections between Scratch place value model and place value in mathematics.  
• Developed connections between time, place value and the Scratch representation of a stopwatch. |
| Variable        | • Used Scratch programming to begin to make sense of variable concept e.g., inside say block, inside if block. | 8.2.2 | • Used Scratch programming to help make sense of variable concept.  
• Used expressions in Scratch to represent a mathematical formula e.g., perimeter.  
• Developed connections between computational representation and algebraic representation of variable in mathematics. |
| Geometry        | • ‘Missed’ key concept of beetle turning through exterior angle. | 8.2.3 | • Moved between representations of geometric ideas in programming on screen and on the floor.  
• Developed connections between geometric structure of a polygon in mathematics and in programming. |
| Thinking with your maths head on | • No observance of theme. | 8.3.1 | • Interpreted scripts step by step to encourage mathematical reasoning.  
• Used scripts/blocks to explain thinking. [Using Scratch as a language.]  
• Developed Thinking with your Maths/Scratch head on. |
| Thinking like a sprite | • No observance of theme. | 8.3.2  
8.3.2.1  
8.3.2.2 | • Used the beetle as an object to think with to draw polygons.  
• Used Post-it Notes and the beetle as an object to think with for variable. |
| Pedagogic practices | • Limited use of technology orchestrations.  
• Variation in engagement with and use of the SM curriculum materials [D2].  
• Little to no use of the 5E’s [D1].  
• ‘Chaotic’ classrooms [D3].  
• Teacher owns the classroom talk, low child agency [D4].  
• Children’s reasoning is not actively surfaced. Teacher provides corrective feedback [D5]. | 8.4.1  
8.4.2 | • Used Scratch live and debugged live.  
• Used Discuss-the-screen and Explain-the-screen orchestrations.  
• Frequently used Multi-modal-representation orchestration beyond the opportunities described in the SM curriculum.  
• Frequently used the 5 E’s to make connections between Scratch and mathematics [D1].  
• Well managed classrooms. [D3].  
• High level of engagement with SM curriculum, adapted some materials [D2].  
• Teacher allows children to explain some of their thinking [D4].  
• Consistent use of formative assessment to surface children’s reasoning. [D5]. |
Chapter 9   Findings and discussion

9.1   Introduction

This chapter provides an overview of my research study and then presents the key findings which resulted from a comparison and analysis of the case studies and augmenting teacher episodes discussed in the previous four chapters. I discuss each of the key findings in further detail in relation to my research questions and to relevant literature.

9.2   An overview of the research study

My doctoral study has adopted a multiple-case study approach augmented with additional teacher episodes to research the evolution of primary school teachers’ knowledge in their engagement with computer-based microworlds that formed part of the ScratchMaths (SM) curriculum. More specifically, what mathematical and computing knowledge is required to effectively teach SM activities on place value, variable and angle to children aged nine to eleven years old. I approached the methodological design by selecting the main theoretical framework of (Mathematical) Pedagogical and Technology Knowledge (MPTK), a dominant model in the field of researching technology with mathematics to examine teachers’ knowledge of using programming to explore and teach mathematical ideas. I extended the framework to include the knowledge domain of Computational Content Knowledge and adapted the framework to show the intersection of mathematical and computational knowledge for teaching because of this addition. The development of the conceptual framework is explained in Chapter 2, section 2.4. I collected research data of the thirteen teachers using multiple methods: video-recorded classroom observations and audio recorded semi-structured and video-recorded task based post-lesson interviews. I identified themes through the data analysis procedure which I used to structure and present the four case studies. Three of the four case studies illustrate and
exemplify individual teachers’ MPTK through rich descriptions and analysis of the data. The final case study, the mathematical sketches, includes rich descriptions and analysis of the data from ten additional teachers who were observed less frequently. Comparative analysis of the data presented in the four cases led to the key findings presented in the next section of this chapter.

9.3 **The key findings**

1. Teaching and building place value models in the ScratchMaths Place value supported teachers to examine their mathematical knowledge for teaching place value and to develop new knowledge connections:
   - the role of the Scratch representation of place value to make implicit knowledge about place value explicit,
   - the importance of the role of 10 to explain and define place value structure,
   - the generality of the structure of the place value system,
   - place value can be generalised to time, and considered as an example of a place value system,
   - the concept of *base ten* is not made explicit when teaching and explaining place value.

2. Teaching using the ScratchMaths Fireworks microworld supported teachers to develop their knowledge of variable within and between contexts. Some teacher’s knowledge of variable concepts in mathematics and in computing has been shown to display the same misconceptions displayed in school children (Brown and Bergman, 2013; Wilkie, 2014). Teaching variable concepts with the Fireworks microworld enabled teachers to open a window on their existing knowledge of variable. By making that knowledge explicit they were provided with an effective bridge between the computing and mathematical contexts to develop that knowledge further.

3. Teaching using the ScratchMaths Fireworks microworld provided teachers with opportunities to confront and reorganise their mathematical knowledge of angle:
9.3 The key findings

- the role of exterior or interior angle in geometrical drawings,
- the prevalence of statements about angle in which the angle type is invisible e.g., “the angles of an equilateral triangle are 60 degrees”, “all the angles add up to 180 degrees”.

4. Teachers developed similes such as thinking like a sprite or thinking with their Scratch/Maths head on as a way to effectively bridge between mathematical and computational ideas. For example, Rina developed and used the phrase “thinking like a sprite” when she taught expressions in mathematics to bridge between variable in computing represented as the Answer block and as words or algebraic letters to represent a variable quantity in mathematics.

5. Teachers used multimodal non-computer-based representations to effectively bridge between mathematical and computing contexts:
   - using the digit cards to represent a sprite and its costumes in computing and to represent a digit in place value in mathematics,
   - using a cut-out beetle on the board to represent the beetle’s movement in Scratch,
   - acting and playing beetle to use geometrical knowledge of the body to represent the beetle’s movement in the microworld,
   - using Post-it Notes to represent the process and overwriting of storing a variable,
   - using role play to show how the computer processes instructions,
   - using multi-link cubes to represent the beetle’s repeated movement when creating algorithms to draw polygons in the microworld.

6. Teachers used a variety of computer and non-computer-based whole-class orchestrations. As their MPTK evolved they employed orchestrations which used Scratch live more frequently. Less confident teachers used Scratch live at the introduction at the beginning of the lesson and only in a scripted manner, or in extreme cases not at all. During the whole-class orchestrations:
- the teacher encouraged explanations and discussions of what was shown on the screen. For example, predicting the output of a script before running it, or modifying a script live in the process of debugging,
- the teacher provided opportunities to develop the children’s mathematical thinking and reasoning through their use of questioning,
- the teacher used multiple representations to bridge between computing and mathematics contexts, the live use of Scratch supported the development of the collective instrumental genesis of the teacher and the children.

7. Teachers developed well-managed and structured classrooms as their MPTK evolved. They provided increased opportunities for children to engage with the mathematics through programming content of the lesson by:
   - structuring the lesson with different types of activity: paired and whole-class discussions to support participation from all children,
   - structuring the lesson to include individual and collaborative computer-based activity,
   - structuring the lesson to ensure pupil computer-based task is discussed before and after activities.

8. As teachers’ MPTK evolved, they more frequently used each and all of the 5E’s from the pedagogic framework. They provided opportunities for effective learning of mathematics through programming by encouraging the children:
   - to explore a single block or output of a short algorithm,
   - to explain a block, or output on the screen of a script or algorithm,
   - to exchange and to explain a child’s script with the teacher, or to debug a child’s script collectively,
   - to envisage the output of the script before running it, or to play as the beetle or to play as a digit sprite,
   - to bridge between computing and mathematics contexts through the use of questions, explanations and by connecting together multiple representations as specified in the SM curriculum.
9. Professional development is essential to support teachers’ development of MPTK. Teachers who had little to no PD made limited use of the 5E pedagogical framework and were unable to effectively bridge between computing and mathematics contexts.

The nine findings together are situated within the inter-related components of the teachers’ MPTK. The first three findings are related to teachers’ evolving knowledge which include multiple components of the framework: Mathematical Content Knowledge, Computational Content Knowledge, Mathematical/Computational Knowledge for Teaching. These findings are directly related to research question one of this study: “How is primary teachers’ knowledge around the mathematical concepts of place value, variable, angle, and mathematical thinking shaped and shaped by their engagement with teaching microworlds around these concepts that formed part of the ScratchMaths curriculum?”. The next six findings are related to teachers’ evolving pedagogy, which include Technology instrumental genesis, Pedagogical Knowledge, and Mathematical/Computational Knowledge. These findings relate to research question two: “How do primary teachers use the digital resources in the ScratchMaths curriculum to orchestrate children’s learning?”. In the next two sections I first present each of the findings in more detail and then discuss them holistically from the perspectives of evolving knowledge and evolving pedagogy.

9.4 Findings related to teachers’ evolving knowledge (RQ1)

9.4.1 Teaching and building place value models in ScratchMaths supported teachers to examine their mathematical knowledge for teaching place value and to develop new knowledge connections.

This finding illustrates how the teachers used the Scratch place value microworld to make the mathematical structure of the place value system explicit through i) the naming of the sprites, ii) the available costumes, iii) the position of the sprites, iv) the conditional check if costume # =, and v) the broadcast and when I receive mechanism. These structural properties of the model are baked into the microworld’s representation of place value through the sequence of
activities. However, analysis of the lesson observations and place value task interviews showed how the teachers used the model to make the implicit aspects of place value explicit. For example, Mary, Rina, and most of group B teachers focused on the role of ten within the model and how it represented both a number displayed using two places, and as part of the condition to check for the 10th costume (displayed as 0) to control when an adjacent digit should increase its value by one. This explicit programmatic relationship reinforces the mathematical relationship of adjacent places, mainly the positional property and the decimal property (Houdement and Tempier, 2019; Ross, 1989) discussed in section 2.2.2. When the teachers asked the children to consider what would happen when the model 0009 is increased by one, whether displayed on the screen or using the sprite digit cards, they first made the mathematical relationship explicit and then programmed the explicit relationship into the Scratch place value model.

The implicit and explicit distinction of place value was further exemplified by Laura and Katherine (Group B) in their discussion of the place value items from the task-based interviews. All other teachers were able to identify and describe the algorithm error the children had made performing the two-column addition in the task-based interview, stating “they should have carried two tens not one”. However, only Laura and Katherine were able to begin to explain why this error had occurred. They explained that the algorithms as presented in the task did not provide sufficient data to explain the source of the children’s error which could be that you always carry ten when doing the pencil and paper algorithm. The other teachers’ limited explanations of the children’s reasoning could be explained by the nature of the interview questions which asked “Which answers have the same kind of error? Explain your reasoning” and “If this is the only pupil’s work, which pupil has the ‘best’ understanding of the algorithm?” since both questions do not prompt for the source of the error. However, what is interesting about the teachers’ responses to the subsequent interview question is their explanation of the purpose of building the Scratch place value model.

Some teachers began to articulate the generality of the structures within the place value system as a rationale for teaching using the Scratch model. Matt from Group B explained in his interview how he could use the model as an additional resource in the classroom alongside place value charts to explain place value. Matt details that the message broadcast (for the
adjacent sprite to receive) is the same as the message that is sent when performing addition, but he considers that the “method of message delivery” is different. I would argue that Matt is explaining that the carrying or exchanging mechanism within whole number addition behaves equivalently to the programmatic representation using broadcast and when I receive in Scratch. This finding is not surprising given that this is one of the main design features of the microworld, but it is interesting in comparison to the rest of the teachers, since several incorrectly stated that the broadcast mechanism sent a message to another sprite, rather than announcing the message to all sprites. However, despite this misunderstanding of broadcast, which may indicate a failure to grasp the general features of the model, Rina, Mary and Katherine began to develop new connections in their knowledge with respect to considering time as a type of place value (see Window on place value, sections 5.2.1, 6.2.1, and 8.2.1 respectively).

All teachers who had taught building stopwatch and timers using the place value microworld began to develop connections between the Scratch representations of the different models: place values, stopwatch, timer. The least well-developed connections were made by teachers such as Anthony from group A who had the least amount of professional development and had not taught ScratchMaths in the previous year. Anthony was able to point out some of the structural differences between the two models, such as needing different place names, although he suggested that the places should be (incorrectly), seconds, minutes, and hours. His lesson observation had indicated that he had struggled with teaching Scratch to the class, and using Scratch live, since he played a video at the start of the lesson. However, Matt, Mary and Laura who had more experience and teaching of ScratchMaths in the first year of the intervention explained that they had used the naming of the places in the base ten model, to support with identification of the names of the places in a stop-watch model. Both teachers also recognised the structural features of the place value model as represented in Scratch, i.e., a message needs to be sent when a conditional check is met when the place had reached its maximum value, such as how to deal with 60 seconds in order to make another place increase by one. Both of their explanations demonstrate that they were considering time as a type of place value system. This finding is further exemplified in the analysis of Rina and Katherine’s (Group B) classroom and interview data. In the classroom, Katherine specifically asked the child to consider the
place value of time when considering the purpose of each place, arguably a stronger connection than that of Matt and Mary. Rina’s classroom observations demonstrated the strength of the connections she had forged between models when she created a broken stopwatch for the class to fix. Rina also explained that the time place value models have provided her with a new way of thinking about place value. She stated that you could calculate time using column addition, a concept she had previously considered was impossible saying to the children “you just can’t”, if you made explicit when to carry and thought the relationships and behaviours of the places within the model.

A final point within this finding is that the teachers’ responses to the place value mathematical knowledge for teaching test items consistently mentioned children’s place value knowledge. However only one teacher (Laura) explicitly referenced base ten in their explanations. Base ten was frequently implied by the teachers’ reasoning e.g., “they have only carried ten and not twenty”. But the importance of base and its role in defining the place structure was not made explicit, an important aspect of generalising in developing mathematical thinking (Cuoco, Goldenberg and Mark, 1996). I would argue that the base ten has been internalised, accepted and is implicitly known by the teacher simply as place value is base ten. This internalisation of base is an exemplification of another aspect of place value knowledge that teachers can take for granted in their knowledge as reported by Thanheiser (2009). However, by confronting base within the programming language, the base became explicit and some teachers noticed something that they had not previously noticed in their knowledge. This finding supports the claim made by Cady, Hopkins and Price (2014) that engaging teachers in activities which support them to reconceptualise their understanding of seemingly simple concepts would be an effective way to engage with knowledge they took for granted.

9.4.2 Teaching using the ScratchMaths fireworks microworld supported teachers to develop their knowledge of variable within and between contexts.

Variable is a difficult topic to teach in computing (Hermans et al., 2018) and in mathematics (Brown and Bergman, 2013; Dogbey, 2016) as discussed in the literature review in Chapter 2. Analysis of the teachers’ data provides examples of how programming using the fireworks
microworld made computing and mathematical knowledge of variable explicit, and how representation in Scratch supported teachers to identify limitations in that knowledge and to develop that knowledge further. For example, case study teachers Sally, Rina, Mary, and Matt and Nancy from Group B of the sketches case all exemplified aspects of using and operating on variable in their programming:

- Stored a value in the **answer** block,
- Used the **answer** block as an input to **move** to control the beetle’s movement,
- Operated on **answer** in an expression to control the beetle’s turn, 360/answer,
- Defined and set their own user defined **variables** to store values,
- Used **answer** and variables to control the number of repeats,
- Used **variable** in an expression in a conditional statement.

The bullet points are keys aspects of using variable in computing and developing knowledge about algebraic variable in mathematics (Noss, 1986; Sutherland, 1992) The bullet points contribute to developing a deeper understanding of variable which the teachers demonstrated in aspects of their teaching and in the interview tasks. However, demonstrating the computing content knowledge required to store and operate on two variables simultaneously was problematic for some teachers such as Matt and Sally. ScratchMaths activity 5.1.3 (see section 4.2.2 for more detail) was designed to engage with this idea within the fireworks microworld. The activity requires building a script to draw a regular polygon, based on two inputs, side length and the number of sides. Working on the script using only the **ask** and **answer** blocks will raise a limitation of the **answer** block, in that two values cannot be store simultaneously, the resolution provides a rationale for needing to use variables.

Matt’s progress on the task illustrates the role of the programming language in making the limitations of his knowledge about variable explicit. For example, he asked the number of sides question and represented the angle of turn as an expression i.e., 360/answer and included a second question for side length placed at the start of the script. Instead, he became stuck in adapting the script to use the second **answer**, he tried different sequences of blocks so that the **ask** block was close to the use of the **answer** in the script. In the interview I noticed Matt was increasingly frustrated and suggested that variables might be useful. He chose to ignore the prompt and instead broke apart the script into pieces and explained each. It is not clear why
Matt chose not to use variables, he may have been holding onto a misconception that the `answer` block can hold multiple values at the same time as found in research of novice programmers (Doukakis, Grigoriadou and Tsaganou, 2007). Matt’s decision is also puzzling given that he had taught variables to store base and height in the lesson proceeding the task. Perhaps he was just following the teaching instructions from the ScratchMaths materials to create and use variables without appreciating the rationale for why variables are required. His approach to the task certainly suggests that using the `ask` and `answer` blocks are much more strongly represented in his knowledge of variable, than that of using the `variable` blocks in Scratch. However, what is key in this finding is how the Scratch programming language enabled Matt to make his knowledge about variable explicit and to show him the limitations in that knowledge. During the task he was unable to overcome the limitation but was able to understand a key aspect of using `answer` store a quantity and the limitations of that programming construct. Matt dealt with the limitation by reducing the scope of his program to use one variable and expressed the perimeter for the polygon which had a fixed side length of twenty, exemplifying an aspect of recognising variable as generalised number in algebraic thinking.

Sally’s case also exemplifies how expressing variable in the Scratch programming language threw light on the limitations of her knowledge. But, unlike Matt, enabled her to develop that knowledge. Over a period of six weeks, Sally’s MPTK developed from a) an initial lesson teaching variable when she stopped the activity realising the limitations of her variable knowledge, b) rethinking her teaching approach by using and defining one variable in simple contexts, and finally c) successfully completing the variable task and using two variables to represent two generalised. It is challenging to identify exactly what changed in Sally’s knowledge over this period, but analysis of the lessons and the interviews suggests that the change in Sally’s MPTK is due to the following contributory factors:

- teaching with the programming language identified limitations of Sally’s knowledge of variable when that knowledge was made explicit, as illustrated by her first lesson.
- her personal orientation to technology and wanting to improve her teaching of ScratchMaths. Sally recognised that she had not had ScratchMaths PD and was not aware of ScratchMaths supporting materials and the 5E pedagogical approach.
Personal orientations were found to be a key contributory factor to developing MPTK (Thomas and Palmer, 2014).

- rethinking variable by breaking down the concept into smaller steps e.g., using one variable in different contexts and using these examples to teach with. The lessons Sally taught after the initial lesson demonstrated a refocussing on exploring and explaining with smaller scripts to develop key aspects of variable, as shown in the bulleted list above.

- defining variable in computing and making connections to variable as generalised number in mathematics. A key moment for Sally was recognising the disconnected nature of her knowledge about variable. She described how she felt trying to teach the variable tasks as grabbing on to blades of grass (the pieces of disconnected knowledge) which she explained to try to slow her down as was being dragged through the activity! The definitions supported Sally to make connections about her variable knowledge between contexts.

- explaining her thinking and reasoning during the task based interviews to support making her implicit knowledge explicit.

Mary’s analysis of her MPTK for variable also exemplifies how she was able to use two variables in a program for a general polygon, which used the mathematical formula for perimeter using two generalised numbers, and did not develop a flawed understanding about variable as found in previous research of novice programmers (Sorva, 2012). However, the limitation of her variable knowledge emerged when she considered a definition for variable. Mary found it challenging to define variable in a computing context and was not initially aware that variables exist in mathematical context. Despite being unable to define variable in words, she was able to use variables intuitively in her observed lessons such as when building an expression to calculate and display the perimeter of the polygon (a mathematical relationship) using variables in a Scratch expression. The programming language provided a bridge to support her knowledge development of variable in both the computing and the mathematical contexts.

Variables in mathematics can take on various meanings such as labels, constants, unknowns and generalised numbers (Philipp, 1992), some of which are closely related to variables in
computing (see section 2.2.3). For example, the concrete instantiations of variable in Scratch as an i) answer block to store a value, ii) costume # block to return the current value of a costume, iii) user defined variable blocks, are closely related to using algebraic variables to represent generalised number and then using the algebraic letters to create mathematical expressions. The Scratch representation of the perimeter of the regular polygon is closely related to that of the perimeter formula in mathematics. However, the idea of a variable being over-written in computing is not present in mathematical contexts since the process of *setting* a value is not an explicit step in mathematical calculation. The relationships of variable within and between mathematical and computational contexts were demonstrated by the case data from Rina and from Nancy (group B sketches). Both teachers made the connection between algebraic representation in mathematics and the programmatic representation of variable explicit. Examples included using algebraic letters to annotate diagrams to represent the side length variable inside the polygon program, connecting mathematical formulae and programmatic expressions, or demonstrating mathematical function machines as behaved the same as ask and answer. Rina developed a pedagogic strategy she named “thinking like a sprite” to create a bridge which enabled her to move between the mathematical representation of variable as generalised number and the computational representation of variable to store a value as discussed below in section 9.6.1.

**9.4.3 Teaching using the ScratchMaths fireworks microworld provided teachers with opportunities to confront and reorganise their mathematical knowledge of angle**

Although the ScratchMaths fireworks microworld explicitly focused on exploring mathematical relationships, Turtle Geometry provided the vehicle for exploring the relationships using variables. Consequently, analysis of the case data also illustrated how teaching using the fireworks microworld provided opportunities for teachers to engage with their concept of angle. When the teachers engaged with how the beetle was turning, they were challenged to examine potential assumptions or limitations within their understanding of angle. Mitchelmore and White (2000, p. 219) stated that developing a more “mature abstract angle concept” depends essentially on learning to link together the various angle perspectives in
different angle contexts. The teachers who had not received professional development such as case teacher Sally, and Betty, Trevor and Elizabeth from Group B all demonstrated that they did not consider the perspective of angle as turning (Clements et al., 1996) as represented by the exterior angle in beetle geometry when teaching using the microworld. Each teacher relied on their procedural mathematical knowledge of triangles stating facts that they were familiar with in contexts outside of programming, e.g., the sum of angles in a triangle is 180 degrees, or 60 degrees for a corner of an equilateral triangle. The nature of this knowledge suggests a more static conceptualisation of angle related to the configurational aspects of triangles. These statements are of course true for the interior angles of a triangle in pencil and paper mathematics, but the limitation of thinking about angle as a dynamic turn is made explicit when representing polygons using Scratch. For example, drawing an equilateral triangle using a turn of sixty degrees will result in three sides forming part of a hexagon. In that moment the teacher will either reconsider how the angle is made as the beetle turns or choose to not engage with the surprising result. This finding is equivalent to previous research of children using Logo who failed to relate the Logo turn parameter to the angle formed by the turtle’s path (Hoyles and Sutherland, 1989; Noss, 1987). Sally initially missed angle as dynamic turn as she engaged with the beetle for the first time in the fireworks microworld but overcame it through her use of playing beetle and multi-modal representations discussed in section 9.6.2 below. Other teachers such as Rina, Mary, and all of the teachers from Group B of the sketches all demonstrated angle as dynamic turn in their observed lessons or in their post-lesson discussions, likely because they had taught year 1 of the SM curriculum which engages with this idea repeatedly throughout Modules 1 and 2.

9.5 Discussion and summary on the findings related to teachers evolving knowledge.

In general, the three findings (9.4.1, 9.4.2, 9.4.3) provided insights into how teaching mathematics through programming using the ScratchMaths curriculum materials enabled both the teacher and the researcher to become aware of the teacher’s own mathematical knowledge, the limitations of that knowledge and in some cases the evolution of that knowledge. Analysis
of the teachers’ MPTK, in the specific mathematical areas of place value, variable and angle, through their use of the fireworks and place value microworlds provided exemplification of how the teachers knowledge evolved. For example, building and representing place value models in Scratch provided a language of representation for the teachers to express the mathematical structure, and a language to be able to explain the generality of the places within the place value system. This finding parallels Sherin’s (2001) research which found that programming could be used to replace traditional algebraic notation when representing the fundamental laws within physics, as the programming representation was more dynamic and process-oriented. By representing place value in the Scratch programming language, the teachers’ place value knowledge was made explicit, such as the role of ten, or the relationship of place value to time. Limitations in that knowledge such as the role of base, or the generality of place value systems, provided opportunities for the teachers to make new knowledge connections and for that knowledge to evolve because of the use of the programming language.

Similar exemplification can be seen within the contexts of variable and angle. Importantly, the limitation of that knowledge had not presented itself within the teachers’ day to day mathematics teaching. Teachers were able to teach children about place value or how to perform a pencil paper algorithm for adding multidigit numbers. However, the programming language provided teachers with potentially new ways of explaining place value, time, and explaining multi-digit pencil and paper algorithms.

The findings on the three knowledge areas of place value, variable and angle exemplify the role of the Scratch programming language in opening a window (Noss and Hoyles, 1996) on teachers’ knowledge and how expressing and representing mathematical ideas in programming was key to disrupt teachers’ knowledge which may have been taken for granted. In the context of place value, the findings show how building place value models can provide teachers with opportunities to examine their own knowledge of the place value system, beyond superficial understanding that rely on recitation of the place value of a digit (Cady, Hopkins and Price, 2014; Thanheiser, 2009). Similarly, from the perspective of variable, teaching using the Fireworks microworld provided opportunities for the teachers to consider and reconsider the concept of variable in both the computing and mathematical contexts. Dogbey (2016) suggested that teacher’s mathematical knowledge for teaching variable was unlikely to improve
without changes to a curriculum that supported it. The findings provide evidence to support that the teachers’ knowledge of variable is improved because of their engagement with the ScratchMaths curriculum that delved more deeply into the concept of variable than traditional curriculum without the use of technology. The finding related to teachers’ knowledge of angle are not new and replicates the earlier findings when working with Logo with children in the classroom.

The findings also resonate with an explicit and implicit knowledge distinction (Davies, 2001; Dienes and Perner, 1999) where implicit knowledge can be regarded as knowing something without being conscious of the abstract structure that underlies that knowledge, and explicit knowledge makes the abstract structure explicit. Davies’ and Dienes’ definition of explicit knowledge considers that the knower makes the knowledge explicit by means of a verbal statement. However, the findings described in this section exemplify using the Scratch programming language in addition to verbal statements are required to make the implicit mathematical knowledge explicit.

9.6 Findings related to teachers’ evolving pedagogy (RQ2)

9.6.1 Teachers developed similes such as thinking like a sprite or thinking with their Scratch/Maths head on as a way to effectively bridge between mathematical and computational ideas.

Analysis of the case data lesson observations revealed how teachers effectively bridged between mathematical and computational ideas through their use of similes. This finding initially emerged in the analysis of Rina’s case data as she had used the phrase “Thinking like a sprite” in a mathematics lesson following a ScratchMaths computer lesson. Rina used the simile to support the children to write mathematical expressions for the total cost of a given word problem:

- Tickets for Little Mix concerts cost £25 each, plus a one off booking fee of £6.
The children were asked to *think like a sprite* to help them explain what question the sprite would ask to represent the word problem. Identification of a question “How many tickets would you like?”, enables connections between the computational use of answer as a variable, storing the number of tickets, to the mathematical use of an algebraic letter to represent the quantity of tickets. The children explained their Scratch expression in words using the **answer** block, and the teacher wrote down the equivalent algebraic expression $25T + 6$. In my experience children often struggle to identify the variable and frequently cite the total cost, a misconception that Rina was very familiar with. Her use of a simile was therefore effective in overcoming that misconception since it supported the children to identify the variable by building a bridge between the computing and mathematical contexts. The pedagogical approach of thinking like a sprite shares some similarities to Hermans *et al.* (2018) pedagogical approach which used variable as a label, e.g. $x$ is 5 rather than $x$ contains 4. The answer is 5 for example is reported by Scratch when the answer block is clicked, and the mechanism for storing the value is not made explicit to user as in the case of the box metaphor, the beetle knows what the value *is*, when you ask the beetle. However, the sprite simile extends what the label notion provides, since the sprite object allowed the teacher and children to consider variable in the context of mathematics. Whether use of this simile results in children developing stronger understanding of variable, or other misconceptions is beyond the scope of this study. However, thinking of variable in this way, and when combined with use of Post-It notes suggests evidence that this approach can be used as an effective way to develop understanding of variable by bridging between both mathematical and computational contexts.

Other instances of the teachers using similes to effectively bridge between computing and mathematics emerged within the data as I looked for (Mason, 2002) pedagogical approach. *Thinking like a digit sprite* was used by Mary to bridge between the physical *Flip, Flip, Nudge, Nudge* game which used the place value digit cards, the Scratch representation using the **next costume** block, and the mathematical context of adding one. She also encouraged the children to *think like a digit sprite* as they bridged between nudging in the game, the Scratch representation of **broadcast** and **when I receive**, and carrying/exchanging in place value. Rina used the simile to effectively bridge from the Scratch base ten place value model to building a stopwatch. She focused on the costumes that need to be available for the **ten seconds** sprite,
e.g., digit cards 0 to 5 to represent numbers bigger than 59 seconds and the reason why another sprite was needed to represent the additional place of one minute. Using this pedagogical approach, to support children’s understanding of time is arguably more powerful than simply stating rote facts about place value such as “there are sixty seconds in a minute”.

*Thinking like a beetle* in angle contexts was used by Rina, Mary, Sally, Nancy and Christine when they acted as the beetle to carry out move and turn instructions to bridge between the computational context and the mathematical as represented by drawings on the floor. Adaptations included children instructing other children to play beetle or using a cut out beetle on the board to represent the sprite. All of these above are variations of *Playing turtle* or *Using the turtle as an object to think with* which Papert (1980) explains are an important construct of the Logo programming language. Using the sprite as an object to think with exemplifies and extends this idea as it moves beyond the scope of Turtle Geometry in which Papert originally situated this idea. His notion of body syntonicity in which children’s understanding of the geometry of their own body as they move and turn can be transferred to the programming language, resonates with the embodied notions of mathematics as captured by thinking like a sprite.

Bridging between the computational and the mathematical contexts was further exemplified by the teachers use of the simile *Thinking with your maths or Scratch head on*. Katherine from sketches Group B, Rina and Mary, all used variations of this simile to support children with their explanations of surprising outputs from running a Scratch program e.g., a rectangle drawn with no base or a place value model which does not change after digit nine. Their use of the simile supported children to engage with their mathematical knowledge to explain the output, what would a rectangle with a base of zero look like? This is not a common question to discuss within the mathematics classroom and the teacher could have easily dismissed the beetle drawing a straight line as a limitation of the Scratch stage which can occur when drawing square near the limits of the stage as shown in Figure 9.1 below.
However, by prompting the children to reconsider what would happen outside of Scratch, i.e., *thinking with their maths head on*, helped to make explicit that a rectangle drawn with no base as the base variable was 0 would result in a straight line since the two short sides would be invisible and the two longer sides would overlap.

*Thinking like a sprite* and *Thinking with your Scratch/Maths head on* is more than a pedagogical approach such as ‘thinking like a computer’ often used to introduce children to learning about the sequential steps of an algorithm as in making a cup of tea or creating a jam sandwich because it illustrates only limited characteristics of algorithm such as the importance of order of steps and sometimes the need for precise language (Benton et al., 2018a). This practice is overly simplistic to represent the more nuanced computational practices that develop as programmers engage with concepts as described in Brennan and Resnick’s (2012) framework of computational thinking. By contrast, the teachers’ development of similes in ScratchMaths such as *thinking like a sprite* for place value, variable and angle, embeds aspects of computational thinking using Scratch: namely precise language, ordered steps, attributes of the sprite, alongside specific computational ideas such as variable, as powerful objects to think with about computer programming and also importantly supports bridging to mathematical thinking and mathematical ideas. Whereas, *thinking with your Scratch/Maths head* provides an effective way to switch focus within each context to support the development of mathematical and computational knowledge between contexts.
9.6.2 Teachers used multimodal representations to effectively bridge between computing and mathematics using multiple non-computer-based representations.

Many of the case study teachers used multiple non-computer-based representations to effectively bridge between mathematical and computational ideas. The ScratchMaths curriculum materials suggest making and creating digit card books to represent the sprite’s costumes for use in the Flip, Flip, Nudge, Nudge activities as exemplified in Mary’s lessons. However, other examples developed by the teachers included i) Sally’s use of multilink cubes to represent the beetle moving in a series of steps which are then repeated in polygon algorithms, ii) Nancy and Christine’s use of post it notes to store the answer in a script and make explicit the overwriting behaviour of a second question, iii) playing beetle as discussed in section 9.6.1 above, iv) Nancy’s addition of masking tape to mark out the beetle’s path on the carpet, v) Sally’s use of a cut out beetle to move and turn on the whiteboard and vi) Rina and Sally’s use of flipchart and white board diagrams. The teachers’ use of multiple representations exemplifies Venkat’s (2015) finding that a focus on key representations is an important component to simultaneously support the development of the teachers’ mathematical learning and the development of their mathematics teaching, or more holistically their MPTK. The analysis of the teacher’s whole-class activity types using Drijver’s et al’s (2010; 2013) orchestrations framework revealed that none of the existing types sufficiently captured the type of activity that the teachers engaged with when using the multiple non-computer-based representations. Their activity went beyond that of the catch-all Board-instruction e.g., the traditional one of a teacher teaching in front of the board since it includes making connections to digital technology which Board-instruction by definition does not. The teachers’ activity could be considered a type of Drijvers et al’s Link-screen-board which stresses “the relationship between what happens in the technological environment and how this is represented in the conventional mathematics of paper, book and board.“ (Drijvers et al., 2010, p. 219). However, the didactical configuration which Trouche (2004) defined as the arrangements of artefacts in the environment is significantly different, since the use of physical objects and physical activity goes beyond that of just using a board and computer screen as in in the definition. Trouche stresses that the didactical configuration of an orchestration has a
Findings and discussion

A strong preparatory aspect, in that the didactical configuration needs to be thought of before the lesson and cannot be changed during the lesson. The teachers use of digit flip books and cut out beetles clearly require significant preparation. Arguably, playing beetle, or the use of the pencil and paper diagrams require little pre-preparation and can be used without preparation. However, the differences in this type of whole-class activity from *Link-screen-board* are sufficient to require the definition of an additional whole-class orchestration.

Drijvers *et al.* (2010) global inventory of whole-class orchestration types distinguishes two elements for an instrumental orchestration: a didactic configuration and an exploitation mode defined as the way the teacher decides to exploit a didactical configuration for their didactical intentions e.g., the decisions in how a task is introduced, or the roles of the different artefacts to be played. I present the *Link-multi-modal-representation* orchestration in the same way with a brief description of the two elements of orchestration theory and illustrate the orchestration with a concrete example. The *Link-multi-modal-representation* concerns the teachers whole-class use of non-digital artefacts to represent ideas from the computational and mathematical environments *and* stresses the relationship between the two. A didactical configuration for this orchestration includes access to non-digital artefacts, access to Scratch, projection facilities and a classroom arrangement that allows the students to follow the explanation and see the artefacts and if required, the screen. The teachers’ exploitation mode may use the artefacts themselves to demonstrate and to explain the link between representations or orchestrate children to use the artefacts whilst being directed by the teacher. The teacher does not always use the digital technology. As an example, teachers exploited this orchestration type to link the multiple representations for adding one in mathematics, the *next costume* block in Scratch, the physical digit sprite book and in the language of addition spoken by the teacher. Student thinking and reasoning is brought to the fore much like the other whole-class orchestrations such as *discuss-the-screen* and *explain-the-screen*. This whole-class orchestration is a significant component of teachers’ evolving MPTK in that the use of the orchestration such as playing beetle, or using post its to store variables, or the place value digit cards, supports the teacher at this stage of their instrumentation of Scratch. The orchestration also provides opportunities for the teacher to make their mathematical and computing knowledge explicit and provides a way to effectively bridge between mathematics and computing contexts.
9.6.3 Teachers used a variety of whole-class orchestrations. As their MPTK developed they frequently used orchestrations which involved using Scratch live.

The analysis of the case data led to the categorisation and frequency of the types of whole-class orchestration type as the teachers taught with Scratch using the ScratchMaths curriculum. There are several similarities and differences between the case study teachers and the teachers from the sketches as show in summary Table 9.1 below.
Findings and discussion

<table>
<thead>
<tr>
<th>Whole class orchestrations</th>
<th>Frequency</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Never</td>
<td>Rarely</td>
<td>Sometimes</td>
<td>Frequently</td>
</tr>
<tr>
<td>Technical-demo</td>
<td>Group A</td>
<td>Mary</td>
<td>Sally</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discuss-the-screen</td>
<td>Group A</td>
<td>Group A</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td>(Anthony/Trevor)</td>
<td>Group B</td>
<td>Rina</td>
<td>Sally</td>
<td></td>
</tr>
<tr>
<td>Explain-the-screen</td>
<td>Group A</td>
<td>Sally</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td>(Anthony/Priya/Elizabeth Trevor)</td>
<td>Group A</td>
<td>Rina</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guide-and-explain</td>
<td>Rina</td>
<td></td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sally</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-screen-board</td>
<td>Mary</td>
<td></td>
<td>Rina</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sally</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spot-and-show</td>
<td>Mary</td>
<td></td>
<td>Rina</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sally</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sherpa-at-work</td>
<td>Group A</td>
<td>Rina</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td>Sally</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Priya)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Board-instruction</td>
<td></td>
<td></td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rina</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sally</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Group A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Group B</td>
<td></td>
</tr>
<tr>
<td>Link-multi-modal-representation</td>
<td>Group A</td>
<td>Group A</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group B</td>
<td>(Katherine)</td>
<td>Rina</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Priya)</td>
<td>Rina</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sally</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Group B</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.1 Table to show frequency of each orchestration for all thirteen teachers from the study. Group A (Betty) means Betty from this group only.

Firstly, the similarities: all teachers made frequent use of the *Board-instruction* orchestration which did not involve the use of technology. However, the lack of technology usage tends to obfuscate how technology could be *discussed* in the lesson without actually using the technology. The content of the whole-class discussion episodes was frequently the programming language. The teaching episodes were used at the start of the lesson to review previous lessons’ computer-based activity, short plenaries to review progress between activities, or instructional sequences to explain and discuss the next activity. An example of
whole-class activity which is not a review of learning, or a preparatory discussion for programming is exemplified within Rina’s case chapter. In Rina’s fourth observed lesson [ERE4(Obsv)] used an ‘un-plugged’ activity, which did not require the use of the computer. Rina’s review of this activity featured an episode which involved Rina asking questions to elicit children’s thinking. These types of discussions are sufficiently different from Discuss-the-screen since the technology was not present, although they did use a printout of the screen, but felt sufficiently closer to “the traditional one of the teacher in whole-class teaching in front of the board” (Drijvers et al., 2013, p. 999) as to not require a new orchestration type. However, it is notable that these unplugged type of activity and discussions are perhaps more unique to the context of computer programming compared with using other digital technology for which Drijvers’ framework aims to typify.

Turning the attention to the differences; the shaded part of the table in grey highlights the orchestrations that were sometimes or frequently used within lessons. Note, if a teacher had been observed multiple times and had used one orchestration once in each lesson or had frequently used another orchestration in a single lesson, both would be categorised as frequently using the orchestration. I considered the presence of teachers use of the orchestration to be indicative of sufficiently developed MPTK required to be able to use the orchestration. However, I questioned the quality of each orchestration, since some teachers demonstrated pedagogic practice which would be exemplars of the orchestration, whilst others fit the didactical configuration and the exploitation, but the quality of the performance was limited. The limitation of only categorising lesson episodes using the whole-class-orchestrations as discussed in Chapter 4 methodology and minimised through application of the TRU framework

Rina, Mary and the group B teachers all taught or supported ScratchMaths teaching in Year 5, the first year of the intervention and therefore had the longest time to develop their MPTK. Looking through this lens, we can see that these teachers made frequent use of at least two orchestrations which involved live use of the technology: Discuss-the-screen, Explain-the-screen, Link-screen-board, Guide-and-explain, Spot-and-show and Sherpa-at-work. All of these orchestrations provided opportunities to develop thinking and reasoning as well as supporting the development of instrumental genesis for both the children and teacher, i.e., collective instrumental genesis. This is not a surprising finding since a teacher using the
ScratchMaths materials as designed, they would consistently be engaging with the 5E pedagogical framework, which embeds live use of programming. For example, two of the 5 E’s, Explore - to explore the outcome of clicking an individual block or experimenting with different inputs, and Explain - to explain scripts block by block to check for understanding of the script, are embedded within every investigation. Consequently, effective whole-class demonstrations or discussions task as presented in the PowerPoint materials require using Scratch live,

As can be seen from Table 9.1 above, those teachers who had received minimal professional development and had limited experience teaching with technology, i.e., Group A teacher are not found in the shaded area of table, made minimal use of Scratch live during their observed lesson. All teachers from Group A, other than Anthony and Trevor who did not use technology at all, did use Discuss-the-screen, but their use of technology live was limited. For example, they tried to discuss what was happening on the screen, but their discussions were largely limited in scope as they could not respond or debug unexpected questions which resulted from the children’s work on tasks

The comparison of the two groups of teachers therefore provides evidence to support the finding that as the teachers MPTK develops so does their use of orchestrations which involve using Scratch live. As can also be seen from summary Table 9.1 (see also discussion in section 8.4.1) teachers with well-developed MPTK are more likely to use Link-multi-modal-representation to bridge effectively between the mathematical and computational contexts and vice versa. The one notable exception to these generalisations is Sally who also had not taught Year 1 of the SM curriculum or received any PD, however her lesson observations demonstrated the orchestrations: Sherpa-at-work, Technical-demo. This is likely a causality of Sally’s prior experience of teaching with technology and specifically with Scratch, providing sufficient PTK to teach using the SM materials. However, as illustrated in Sally’s case, when she became aware of the 5E pedagogical framework and the SM teaching materials, Sally evolved her pedagogical approaches through use of the Link-multi-modal-representation to effectively bridge between computing and mathematics.
9.6.4 Teachers created well-managed and structured classrooms to provide opportunities for children to effectively learn mathematics through programming.

The findings in this and the following section emerged from analysis of the teachers’ lesson observation data using the adapted Teaching for Robust Understanding (TRU) framework (Schoenfeld, 2013; Schoenfeld, 2014b) explained in Chapter 3. Within each dimension, a lesson is broken down into ten minute teaching practice, each of which can be rated on three levels of sophistication. For example, in dimension 3 *Equitable Access to content*, the practice is rated as *basic* if “there is differential access to or participation in the mathematical content, and no apparent efforts to address this issue” (Schoenfeld, 2018, p. 493). They are rated as *proficient* if “there is uneven access or participation, although there are some clear efforts by the teacher or in the materials to provide mathematics access to a wide range of students” (ibid). A rating of *distinguished* is given when classroom activities support meaningful mathematical participation.

Looking across all the case data, and at the dimensions of activity, the most notable is the consistent high scoring of Dimension 3 - *Equitable Access to content*. Mary, Rina and all of the group B teachers demonstrated a distinguished level of sophistication in managing their classroom activities in all observed lessons as shown by the data presented in i) Chapter 5 Mary, section 5.4.4, ii) Chapter 6 Rina, section 6.4.4 and iv) Chapter 8 Group B, Sketches, section 8.4.2. Their lessons featured the following characteristics which enabled students to have an equitable access to content:

- Lesson structured to provide multiple activity types: whole-class-discussion, whole-class discussion using computer, paired pupil discussion, individual computer work, unplugged activities, paired computer work.
- Timed activities, for example “Thirty seconds to talk to your partner”, 15-20 minutes of pupil computer work followed by whole-class discussion,
- Use of well-established assessment for learning routines to ensure participation from all children, e.g., cold calling children, selecting a lolly stick to randomly select a child.
Findings and discussion

- Use of well-established behaviour for learning routines for using a computer, e.g., lids closed, hands off keyboards etc.

The ScratchMaths curriculum provided content which features discussion questions, activities to foster collaborative working, all underpinned by the 5E pedagogical framework and designed to be accessible to all children. However, what is key in this finding, is the implementation of the curriculum by the teacher as demonstrated by the teachers as they overlaid their own established pedagogical approaches to ensure equitable access to the content for all children. The teachers’ holistic development of MPTK to enable equitable access can be argued to be a consequence of teaching the Scratch Maths curriculum for the previous year. This suggestion is strengthened when comparing lessons from those teachers from the Group A of the snapshot case. The teachers with little experience of teaching ScratchMaths or with teaching with technology featured a lesson which was more chaotic. The children’s use of the computer was not managed in any obvious way: a task was set which the children just got on with. The teacher chose to hide from providing opportunities to discuss the tasks in depth at a whole-class level, or to discuss as a whole-class when an issue arose. Instead, the teacher chose to provide the children longer opportunities to work at the computer and provided some 1-1 support. The impact was a considerable off-task behaviour and limited opportunities to develop their programming or make connections to mathematics. Discussing the lesson with the teachers confirmed their limited MPTK such as missing the key idea of exterior angle, or lacking confidence and to use Scratch live.

Priya from Group A and Sally are notable exceptions and suggest that with continued use of the ScratchMaths curriculum and teaching using well-established pedagogical practices for managing behaviour that the lessons can provide equitable access to the content. Sally’s first observed lesson tended to privilege those children that were having success rather than engaging all children. Sally explained that she was unaware of the teaching materials and had planned her own lesson based on creating a final Fireworks scratch project without breaking down the task into more manageable chunks. For Sally this was a key learning that she demonstrated in her subsequent lessons by carefully where she created more focused tasks broken down into steps and created discussion opportunities before and after the task. Her change in approach was facilitated by the SM curriculum materials and supported by engaging
with the 5E pedagogical approach, both which she was unfamiliar with due to lack of professional development. Priya, exemplified a teacher who had received the PD, had not taught the materials in the previous year, but had engaged with the teaching materials. Although only a single lesson, she demonstrated that she had tried out activities before the lesson and prepared additional notes to support her lesson. Her lesson was consequently as well structured and well-structured as the more experienced teachers. However, these additional aspects of teacher lesson preparation were not present in Elizabeth, Trevor’s, Anthony or Betty’s lessons. This finding suggests that in order to teach lessons which enable equitable access to mathematical content, requires both carefully designed curriculum materials and a teacher who can engage with the pedagogical approaches which underpin the curriculum and also facilitate classroom activity using their own well-established pedagogical approaches to manage learning.

9.6.5 **Teachers who frequently used of each and all of the 5E’s from the pedagogic framework, provided opportunities for effective learning of mathematics through programming.**

This second pedagogical finding exemplified by analysis of the case data using the adapted TRU framework is related to dimension 1 *Mathematics Through Programming*, i.e., the extent to which the teacher supports exploration and development of mathematical ideas through learning to program. Rina demonstrated the highest level of sophistication within her classroom practice through her consistent use of each of the 5E’s across the lessons observed. What is notable about her practice is that within each lesson she used most or all of the 5E’s and used them multiple times. By analysing 10 minute episodes the weighted average represents her use of orchestrations and the 5E’s within the lesson holistically. For example, in one episode she started by asking the children to *Explore* the output of a broken timer script, and then asked the children to *Explain* why the script was broken. In the same episode, she encouraged the children to predict what would happen to the script after a change had been made before running the script, exemplifying *Envisage*. The children *Exchanged* ideas in pairs as they worked on debugging the task and within the whole-class discussion Rina *bridged* to mathematics contexts discussion how the timer should operate in the context of time as a type of place value. In
another episode she made use of *Explain* and *bridge* to mathematics but did little exploration. However, across the lessons she frequently used each and all of the 5E’s.

Other teachers who had also taught ScratchMaths in year 1, made consistent use of *Explore* and *Explain* and *Envisage* and sometimes *Exchange* but made less frequent explicit links to mathematics (bridge) within every lesson. Of those that did use bridge frequently within a lesson, such as Nancy, the teacher used multi-modal representations as discussed in finding 9.6.2 above. In the post-lesson interviews many teachers such as Mary, Sally, Matt, Katherine, recognised that they needed to make greater links to mathematics, but for them the hurdle was frequently their technical knowledge of using Scratch. Looking more deeply at Sally’s TRU data over time demonstrated highlights this point. As Sally became more confident with teaching Scratch using the ScratchMaths materials, so did her proficiently in this dimension. She frequently used multi modal representations to bridge effectively between mathematics and computing.

### 9.6.6 Professional development is essential to support teachers’ development of MPTK.

The relationship of the teachers’ uses of the 5E’s to the teachers’ evolving MPTK is demonstrated by those teachers who had not taught ScratchMaths in year one or had received little professional development. Their lessons (other than Priya) were characterised by little or no use of the 5E pedagogical framework. However, when the lesson did feature a whole-class technology orchestration such as *Discuss-the screen* but without the presence of the 5E’s the discussion was largely unstructured and involved the teacher telling the children to do. For example, telling the children to follow precise instructions to create a script live using Scratch. The lesson may have resulted in the children having success with creating a working program in Scratch but the opportunities for robust understanding in mathematics and computing were limited.

The teachers who had received PD in both years of the intervention and had taught ScratchMaths for a year had stronger MPTK than the Group A teachers who had not. PD is essential to provide teachers with pedagogical approaches to teach mathematics through
programming. The careful design and sequencing of the ScratchMaths can only go so far in providing activities that engage children in learning mathematical and computing concepts. The role of the teacher was essential in creating effective bridges to connect computing and mathematical ideas. When a teacher had strong MPTK they had good mathematical content knowledge, good understanding of how to use programming and positive attitudes towards the classroom use. The teachers with lower MPTK relied on teaching the what of an activity rather than the why and their emphasis was on the use of technology rather than the mathematics itself.

9.7 Discussion and summary of the findings related to teachers evolving pedagogy.

The theory of instrumental orchestration provides insights to answer the question of how the teacher fine tunes the student’s use of technology in order to guide the children’s instrumental genesis. Using Drijver’s (Drijvers et al., 2010; Drijvers et al., 2013) whole-class activity orchestrations framework provided an analysis of the type and frequency of activities which teachers engaged with as they taught with Scratch and ScratchMaths curriculum materials. The data analysis provided insights into the teacher’s evolving pedagogy as they learned to use technology to teach mathematical concepts. The findings exemplify how the teacher used the whole-class orchestrations to manage the individual instruments in the collective learning process and thus enhance the collective instrumental genesis. Examination of the frequency of the whole-class activity types suggests that as teachers develop their MPTK they are more likely to use the technology specific orchestrations, to program or to debug programs live. This finding is in line with research conducted by Simsek (2021), who found that expert teachers routinely made use of the dynamic mathematical technology system live during whole-class activities and when working with individual students.

The analysis of the whole-class orchestration data led to the identification of a specific type of activity unique to the teaching of mathematics through programming that was not included in Drijvers et al. (2013) typology. The activity included the teacher’s use of multiple non-digital representations such as the digit cards, or a cut-out beetle, or ‘playing beetle’ to bridge between
the computing and mathematics environments. This type of activity goes beyond Drijvers et al.’s catch all Board-instruction e.g., the traditional one of a teacher in whole-class teaching in front of the board since it does include making connections to digital technology. The teachers’ activity could be considered a type of Link-screen-board however the didactical configuration was different, since the use of physical objects, or physical activity goes beyond that of a board and computer screen. Likewise, the teachers choice to not use technology, despite it being present, shares the characteristics of Tabach’s (2011) Not-Use-Tech orchestration, with an important difference. The Scratch programming language is one that can be spoken or written down without needing the use of technology such as in the use of unplugged activities which had been built into the ScratchMaths curriculum. This is significantly different than the teacher not choosing not to use technology despite it being present.

A second significant difference in the whole-class activity compared to Link-screen-board or Not-Use-Tech is the teachers’ exploitation mode. The teacher chose to use additional representations, rather than teaching without the use of technology present to support the bridging between mathematical and computational ideas. The representations as used by the teacher enabled the teacher to connect ideas from both mathematics and computing, such as the use of the digit cards to represent the sprite’s costume and its place value digit and make connections to the language of Scratch which could (though not always) be displayed on the screen. The linking of multiple representations of constructs was also found by Thomas and Palmer (2014) for teachers with high MPTK, however the constructs were more mathematically similar, for example, graphical, tabular, or algebraic. Whereas the teachers use of additional representations in the ScratchMaths lessons were dynamic and not typical mathematical representations of constructs. I therefore characterized this as an additional orchestration type as Link-multi-modal-representation as exemplified in the context of Scratch programming. Whether this orchestration characterises teacher’s activity in contexts outside of Scratch programming, but still within the domain of using technology to learn mathematics is beyond the scope of this thesis but warrants further research.

Deeper analysis of how the teacher performed each of the whole-class activity types, i.e. the didactical performance (Drijvers et al., 2009) considered the questions posed and how the teacher dealt with aspects of the mathematical task or technological tool. The analysis revealed
further approaches teachers developed to effectively bridge between the mathematical and computational contexts. The findings across the cases provide evidence of the teachers developing their own similes to effectively bridge from mathematics to computing and from computing to mathematics. The use of analogies are one mechanism that can facilitate transfer based on existing research on programming and mathematic (Fogarty, Perkins and Barell, 1992; Salomon and Perkins, 1989). However, my findings exemplify the teacher’s use of similes and how they used them in context of teaching mathematics through programming. For example, the development of the simile Thinking like a sprite to support the children with forming algebraic expressing in mathematics. The teacher used the sprite from the ScratchMaths microworld and the computing ideas of Ask and Answer to effectively bridge to the algebraic context sing words and symbols. The development of these similes are an expansion and exemplification of Papert’s ideas of object to think with and playing turtle. As Papert argued, using the turtle as an object to think with, and as my findings also demonstrate the teachers use of similes in creation of objects to think with can be used to make some mathematical ideas more learnable in the context of programming.

Analysis of the individual lessons using an adapted version of the Teaching for Robust Understanding (TRU) framework provided a complementary lens to analyse the lessons holistically and addressed how the aspects of the MPTK framework are inter-related which cannot necessarily be analysed discretely within the lesson. For example, the 5E pedagogical framework required the teacher to have knowledge of computing concepts and of Scratch when using Explain or responding to children’s explanations. Similarly, bridging required the teacher to have knowledge of computing and of mathematics and to know how the two are inter-related. The importance of well managed and structured classrooms would give children opportunities to develop their authority and agency, alongside opportunities for the children to explain their thinking. A key pedagogical finding was that in order to provide an effective classroom to learn mathematics through programming the classroom must be well managed and structured. This is not surprising, given that chaotic classrooms are unlikely to yield effective teaching and learning! However, it is important to state what the specific features of the classroom look and sound like.
The most effective lessons with respect to providing opportunities for children’s mathematical understanding were classrooms that featured frequent use of each and all of the 5E’s. This finding could be argued as a causation of how I adapted Dimension 1 of the framework since greater use of the 5E’s demonstrates a greater fidelity of the implementation of the ScratchMaths curriculum materials. However, other aspects of the framework, such as the extent to which the teacher provides the children with agency and ownership (Dimension 4) of the mathematics in their explanations of their thinking or the teacher’s use of formative assessment (Dimension 5), are all inter-connected with the 5E’s and did not require any adaptation to the TRU framework.
Chapter 10  Conclusion

The broader aim of this study was to explore the evolution of teachers’ mathematical and pedagogical knowledge as they engaged with computer-based microworlds that formed part of ScratchMaths. ScratchMaths is a two-year mathematics and computing curriculum designed for pupils aged nine to eleven years old. This final chapter discusses the significance of the research findings and the contribution this study makes to research in the field of teacher knowledge for teaching mathematics through computer programming. Following this, the limitations of the findings are discussed. The chapter concludes by considering the implications of this study for future research.

10.1 Introduction

The thesis set out to trace the evolution of teachers’ knowledge as they taught mathematics through programming and how mathematical ideas mediate and are mediated by the ScratchMaths curriculum. Chapter 1 stated that researchers had struggled to agree whether programming benefits mathematical understanding. This position was based on prior research into using Logo programming to learn mathematics. The research had highlighted the benefits of using Logo programming to engage with learning some mathematical concepts such as learning about geometrical concepts (Clements, Battista and Sarama, 2001; Yelland, 1995), or children’s understanding of algebra related concepts (Noss, 1986; Noss, 1987; Sutherland, 1987; Sutherland, 1992). However, the early research also raised the critical importance of the teachers’ capacity to make connections between the programming and the mathematical concepts at stake. The chapter then discussed the policy change in England in 2013 with the introduction of a new National Computing Programming for children of all ages. The curriculum was a significant change and replaced the previous curriculum which only focused on using ICT software applications, such as word processing or spreadsheets. The purpose of
the new curriculum is to ensure that children engaged with core ideas from computer science, and had the opportunity to put their knowledge and understanding to use through programming (Department for Education, 2013b). As a response to support teachers with the development of the skills required to implement the new computing curriculum, the ScratchMaths research project was created. For three years, I collaborated as part of a cross-discipline team to develop a two-year mathematics and computing curriculum called ScratchMaths (SM) for Year 5 and Year 6 children. The SM team comprised of researchers who had researched extensively with Logo and its variants, as well as specialists in computer science, design research, mathematics pedagogy and teacher development. The curriculum consisted of a comprehensive set of materials designed to promote the teaching of carefully selected core ideas of computer programming alongside specific fundamental mathematical concepts. The curriculum was designed to be accessible for all children. SM also devised extensive teacher support materials and professional development to support the teachers with developing their knowledge to implement a brand-new curriculum. The intervention was implemented as a randomized control trial and evaluated by a research team appointed by the Education Endowment Foundation (EEF) whose measurement of success of the intervention was children’s mathematical attainment in the Key Stage 2 assessment at the end of Year 6.

The EEF’s evaluative major focus on pupil attainment provided a research opportunity for this work to focus on the teachers and their development, thus providing a more complete story of the SM intervention. The concept of computer programming was new to most primary school teachers at the time of SM and they were unlikely be confident in skills and knowledge to teach such a computing curriculum even having engaged in professional development. Collecting data during the SM intervention provided a unique opportunity to examine how teachers’ knowledge evolved as they engaged with learning to computer program and actually engage in the teaching of mathematical concepts through programming. Thus, the broad focus of this research was an exploration of the teacher knowledge required to teach a curriculum at the intersection of mathematics and computing which gave rise to two research questions:

- (RQ1) How does primary teachers’ knowledge and mathematical thinking of selected mathematical concepts shape their engagement with teaching microworlds around
these concepts, and reciprocally, how is their knowledge and thinking shaped by this engagement?

- (RQ2) How do primary teachers use the digital resources of the ScratchMaths curriculum in their teaching to orchestrate children’s learning?

This study adopted a multiple-case study approach augmented with additional teacher episodes to investigate teachers’ Mathematical Pedagogical Technology Knowledge (MPTK) as they taught mathematics through programming using the SM curriculum materials. A qualitative approach was taken to address both research questions using a mixed methods approach consisting of lesson observations, ‘think aloud’ while engaging with computer-based tasks, and post-lesson semi-structured interviews.

### 10.2 Implication of findings

The findings in Chapter 9 related to RQ1 raise important implications for the role of computer programming in making teachers’ mathematical knowledge explicit and how computer programming can shape that knowledge for teaching. The review of the literature in Chapter 2 found researchers exploring teachers’ mathematical knowledge in the contexts of place value and variable and questioning the limitations of that knowledge. Concerns (Thanheiser, 2009; Thanheiser, 2015) had been raised about teachers’ conceptions of multidigit whole numbers and how their familiarity with the base ten system which could lead to a superficial understanding of the system which relied only on recitation of place value rules. Similarly, researchers examined teachers’ conceptions of variable (Brown and Bergman, 2013; Girit Yildiz and Akyuz, 2019) and found that many of the misconceptions displayed by children were also present in teachers’ knowledge. Both fields of education called for research that would enable teachers to engage in activities which would enable them to confront and then to reorganise their knowledge.

Firstly, in relation to place value, expressing place value using the Scratch programming language provided some teachers opportunities to see place value as more than just a written algorithm to be performed or as place value facts such as the names of places. The teachers used the SM place value microworld to re-express place value as a model which had a
mathematical structure and operated on defined rules. A consequence of this representation is the light it threw on teachers’ knowledge of time. Some teachers developed connections within their knowledge and expressed time as type of a place value system. I recall making the same connection during activity design sessions with Ivan Kalas as he worked on the computing aspects of activities and I developed the mathematical. However, what I had not expected from the teachers’ engagement with the place value microworld was how the role of base was taken for granted, or more specifically how the base was invisible. Most teachers were able to explain the role of ten in the Scratch place value models but did not explain how ten was an important component of the base of the place value mathematical system. However, engagement with the SM place value microworld made the role of ten explicit for some teachers and shaped their thinking when considering the base system for time. This finding is particularly notable since the compulsory national mathematics curriculum (Department for Education, 2013b) for Key Stage 1 and 2 in England does not make any explicit reference to base ten in any of the statutory programmes of study and does not require the teaching of other place value systems. International curricula as in the United States, does make the base explicit in all curriculum statements. The implications of this omission on learners in the English primary curriculum are beyond the scope of this study but could be of interest for future research.

Secondly, teacher’s mathematical knowledge in the context of variable evolved because of first expressing variable in the programming context and then bridging to the mathematical context. The teachers’ bridging between the contexts was an important mechanism in the evolution of that knowledge of variable in both contexts and discussed further in the context of RQ2 below. However, together the findings related to RQ1 suggest that that a curriculum which explores mathematical ideas through programming such as SM does not just provide activities for children to engage with as they learn to computer program but simultaneously also provides opportunities for teachers to confront and possibly reconceptualise their mathematical knowledge for teaching place value and variable.

The findings in Chapter 9 related to RQ2 exemplify how teachers effectively bridged between computing and mathematics contexts when teaching mathematics through computer programming. Previous research had highlighted the importance of designing bridging activities that “consolidate, support and sustain students’ mathematical way of thinking beyond
their digital experience” (Geraniou and Mavrikis, 2015, p. 321) in the mathematics context, and the importance of unplugged activities (Bell and Vahrenhold, 2018) in computer science to communicate concepts in an engaging and lasting way. The findings presented in the thesis exemplify the pedagogical approaches teachers developed to bridge between computing and mathematics, and how they can be used to make connections between the mathematical and computing ideas. Key to this finding was the teachers use and linking together of multiple representations outside of the programming environment that represent mathematical ideas and computational ideas from inside the programming environment. This finding resonates with Morgan and Kynigos (2014) analysis of different modes of representation within a digital environment, and the importance of interacting with a digital environment’s multiple representations, to enable students to access the “deep structure” of the system. Of course, the representations discussed here differ as they exist both inside the computer as the programming language representation, as programming blocks and as attributes of the sprite such as costumes or movement, and outside of the digital environment such as the teachers use of playing beetle, or role play, or a cut out beetle. This finding led to the identification of new type of whole-class orchestration referred to as Link-multi-modal-representation which teachers used to represent ideas from the computational and mathematical environments and to stress the relationship between the two. The orchestration shares the same features of Drijvers et al. (2010) Discuss-the-screen and Explain-the-screen to provide opportunities for students to explain their reasoning and thinking but adds the use of additional representations outside of the computer. Another pedagogical approach was the teachers’ use of similes to effectively bridge between computing and mathematics contexts. The significance of this finding further exemplifies Papert’s (1980), “Playing turtle” in his definition of body syntonic learning or more recently discussed in the literature as embodied cognition (Lakoff and Núñez, 2000). The teachers use of Thinking Like A Sprite and Thinking with your Scratch or Maths head on also exemplifies in a metaphorical sense the programming language as an object to think with.

A final implication of the findings is the essential role of professional development in supporting teachers to develop the knowledge required to be able to effectively teach a curriculum which explores mathematics through programming. Any professional development must go beyond just learning how to program and also include how to operationalise the
curriculum. The existence of teacher curriculum materials is not sufficient to meet this aim. Brown and Campione (1996) used the term *lethal mutation* to describe the problem where teachers do not understand the underlying (intervention) principles such as the pedagogic framework but instead focus on the surface features, such as doing the activities. Lethal mutations were clearly evidenced within the case studies and the additional teacher episodes for teachers who had limited professional development and attempted to teach using the SM curriculum materials. The power point presentations had made explicit links within the materials to the 5Es framework (Benton et al., 2017). However, this was not sufficient when some teachers were unaware of the supporting teacher materials or the purpose of the 5E pedagogical framework. The consequences of the lethal mutation were exemplified as the teacher not understanding a core of idea of *turn* from the module, the teacher not understanding how variable in computing can be used to bridge to variable in mathematics and some teachers providing limited opportunities for children to develop their knowledge of programming and to make connections to mathematics. The significance of this finding is for curriculum designers (and researchers) to consider the essential role of professional development if they are to truly reach the transformational goal of the National Computing Curriculum to support teaching and learning in every subject (Jones, 2015).

### 10.3 Contribution to literature and methodology

The PhD research contributed to the existing literature and methodology in three ways i) developing an understanding of how teachers’ mathematical pedagogical technology knowledge (MPTK) is mediated and mediated by teaching mathematics through programming as they bridge between computing and mathematics contexts, ii) specifying a theoretical model for teachers’ knowledge for teaching mathematics through programming in the primary setting and iii) adapting the Teaching for Robust Understanding framework in the context of teaching mathematics through programming in the primary setting.

**Contribution 1a:** The role of computer programming in making teacher’s mathematical knowledge explicit and how expressing that knowledge in programming can (re)shape that knowledge for teaching.
The study has specifically focused on examining teachers’ evolving MPTK as they engaged with teaching the mathematical topics of place value, variable and angle through Scratch programming. The study has exemplified how the knowledge required to teach at the intersection of programming and mathematics goes beyond knowledge in each domain. Some teachers created new connections in their knowledge as they bridged between the mathematics and computing contexts. Other teachers showed limitations in their mathematical knowledge of place value, variable and angle which impacted their capacity to bridge between the two contexts.

**Contribution 1b:** The pedagogical approaches teachers use to bridge effectively between computing and mathematics contexts.

The study has illustrated how primary teachers used the digital resources from the ScratchMaths curriculum to bridge effectively between computing and mathematics contexts. The most effective lessons featured teachers’ use of all of the 5Es from the 5E pedagogical framework. Previous research has highlighted the importance of bridging activities to support students to develop a mathematical way of thinking beyond their digital experience. This study extends that research by exemplifying the pedagogical approaches teachers used to bridge between programming and mathematics. Key to the literature contribution is the identification of a whole-class orchestration activity type *Link-multi-modal-representation* which teachers used to link together ideas outside of the programming environment that represented mathematical and computational ideas from inside the programming environment. This whole-class activity orchestration type extends Drijvers *et al.* (2013) typology in the context of computer programming. Whether this orchestration type is unique to teaching mathematical through computer programming is beyond the scope of this thesis and a question for future research.

Since this research was completed, there have been few published studies (Calder, 2018; Miller, 2019; Rodríguez-Martínez, González-Calero and Sáez-López, 2020) that focussed on exploring mathematical content through programming in the primary context. The studies provided additional evidence that learning using programming can lead to improved student understanding of some mathematical concepts. However, all of the studies focused on student
activity rather than the role of the teacher and their knowledge. One notable exception was a recent conference paper presented at the 14th International Conference of the Learning Sciences in June 2020. Fofang et al. (2020) explored children programming a spherical to engage with prime and composite numbers. The authors designed activities where mathematics served as a context to employ computational thinking strategies and where the computational thinking helped to deepen mathematical engagement. This curriculum approach closely parallels that of ScratchMaths. Their early findings showed vignettes which exemplified what integrating CT into mathematics activities can look like and focused on student interactions. Although they did not research the teacher, they drew attention in their findings of the essential role of the teacher to draw out and facilitate the mutual supportiveness of mathematics and CT within integrated lessons and the limited research (they cite two studies in science education) available. Contributions 1a, and 1b of this thesis contribute to this developing area of research.

**Contribution 2:** Specifying a theoretical model for analysing teachers’ knowledge for teaching mathematics through programming in the primary setting.

The study has shown how the PTK framework (Thomas and Palmer, 2014) can be extended and used as a key indicator of teacher’s evolving knowledge when teaching mathematics through programming in the primary setting. The model was used as the main theoretical lens to design the whole research study including the data collection and analysis. I developed the model to include an additional component for Computational Content Knowledge and adapted the Mathematical/Computational Knowledge for Teaching component to bring together both the mathematics and computational knowledge with the corresponding pedagogical knowledge. On their own the different components of Thomas and Palmer’s (2014) original PTK framework do not exemplify how the model can be utilised to examine teacher’s knowledge. To address this limitation I used Clark-Wilson and Hoyles (2017) approach of characterising each component of the model to identify what a teacher might enact or describe in a lesson when doing the “mathematical work of teaching” involving the use of mathematical and technological representations (see Table 3.7 in Chapter 4). Analysis of the teacher data as presented in the case studies and the additional teacher episodes showed how the *a priori* characterisations were observed in teachers’ practice in the classroom.
The adapted MPTK framework for teaching mathematics through programming in the primary setting theoretical framework is presented in Figure 10.1 below. The model has the following components:

- **Pedagogical knowledge**: A teacher’s knowledge of the principles and strategies of classroom management.
- **Mathematical content knowledge**: A teacher’s own knowledge of mathematics.
- **Computational content knowledge**: A teacher’s own knowledge of computing and computer programming.
- **Mathematical/Computational Knowledge for Teaching**: The mathematical and computational knowledge alongside the pedagogical knowledge needed to perform the teaching of mathematics through programming using the microworlds and tasks from the ScratchMaths curriculum.
- **Personal orientations**: The teachers’ goals, attitudes, dispositions, beliefs and their perceptions of the nature of mathematical knowledge and how it should be learned with or without technology.
- **Technology instrumental genesis**: The process by which the teacher’s mathematical and computational knowledge shapes and is shaped by their interactions with Scratch as they accomplish a particular ScratchMaths task. This also incorporates the teacher’s understanding of the development of the children’s processes of instrumental genesis, as they engage with Scratch to explore mathematical ideas.
Contribution 3 Adaptating the Teaching for Robust Understanding (TRU) framework for application to teaching mathematics through programming in the primary setting.

My research has made a methodological contribution to the literature by adapting the TRU framework (Schoenfeld, 2013; Schoenfeld, 2014a; Schoenfeld, 2014b; Schoenfeld, 2018) for equitable and robust learning in mathematics classrooms using programming to explore mathematical ideas. The TRU framework consists of five key dimensions of the learning environment 1) The content, 2) Cognitive Demand, 3) Equitable Access, 4) Agency, Ownership and identity, and 5) Formative assessment. Dimension 1 was adapted for the content of learning mathematics through programming as shown in Table 10.1 below; my contribution is shown in bold black text on blue background (all colours are preserved from Schoenfeld (2018)). The framework was used to analyse 28 lessons across the three teacher case studies.
and episodes from ten additional teachers. Use of the framework provided a robust way to analyse the lesson holistically since it brought together the instrumental orchestration framework, the 5E pedagogical framework and the ScratchMaths curriculum. Although the framework in this research was used to identify the proficiency of each teacher in utilising the ScratchMaths curriculum material and their use of the whole-class activity types, the adapted framework has the potential to be used in future professional development such as in lesson study (Schoenfeld et al., 2019) for learning mathematics through programming.
### Conclusion

<table>
<thead>
<tr>
<th>Dimension 1</th>
<th>Dimension 2</th>
<th>Dimension 3</th>
<th>Dimension 4</th>
<th>Dimension 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics through programming</td>
<td>Cognitive Demand</td>
<td>Equitable access to content</td>
<td>Agency, Ownership and Identity</td>
<td>Formative assessment</td>
</tr>
<tr>
<td><strong>To what extent does the teacher support exploration and development of mathematical ideas through learning to program?</strong></td>
<td><strong>To what extent are students supported in grappling with and making sense of mathematical concepts?</strong></td>
<td><strong>To what extent does the teacher support access to the content of the lesson for all students?</strong></td>
<td><strong>To what extent is students’ mathematical thinking surfaced; to what extent does instruction build on student ideas when potentially valuable or address misunderstandings when they arise?</strong></td>
<td></td>
</tr>
<tr>
<td>Basic (1)</td>
<td>The teacher makes limited or no use of technology, simply tells students what to do with no clear aim in mind. Student computer activity is unstructured, unfocused and with little or no use of the 5E’s.</td>
<td>Classroom activities are structured so that students mostly apply memorized procedures and/or work routine exercises.</td>
<td>There is differential access to or participation in the mathematical content, and no apparent efforts to address this issue.</td>
<td>Student reasoning is not actively surfaced or pursued. Teacher actions are limited to corrective feedback or encouragement.</td>
</tr>
<tr>
<td>Proficient (2)</td>
<td>The teacher uses orchestrations and engages with the 5E pedagogical framework, but connections with mathematics (bridge) are limited. Students are given some, but limited opportunities to Explore, Explain, Envisage and Exchange when working with the computer.</td>
<td>Classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to “scaffold away” the challenges, removing opportunities for productive struggle.</td>
<td>There is uneven access or participation but the teacher makes some efforts to provide mathematical access to a wide range of students.</td>
<td>Students have a chance to explain some of their thinking, but the teacher is the primary driver of conversations and arbiter of correctness. In class discussions, student ideas are not explored or built upon.</td>
</tr>
<tr>
<td>Distinguished (3)</td>
<td>The teacher uses orchestrations and makes meaningful connections between representations in programming and representations in mathematics (bridge).</td>
<td>The teacher’s hints or scaffolds support students in productive struggle in building understandings and engaging in mathematical practices.</td>
<td>The teacher actively supports and to some degree achieves broad and meaningful mathematical participation; OR what appear to be established participation structures result in such engagement.</td>
<td>The teacher solicits student thinking and subsequent instruction responds to those ideas, by building on productive beginnings or addressing emerging misunderstandings.</td>
</tr>
</tbody>
</table>
Table 10.1 Adapted Teaching for Robust Understanding (TRU) summary (Schoenfeld, 2014a; Schoenfeld, 2018) scoring rubric used for classroom analysis of Teaching Mathematics through programming
10.4 Limitations of the research design

The teacher sample upon which the findings are based were selected to be a representative sample of those who had taught the second year of the ScratchMaths intervention. In addition, the schools that had opted to be part of the ScratchMaths intervention could be considered biased since they had a positive orientation to using and trialling new technology in the classroom. Personal orientation is an important contributing factor to the development of teachers’ MPTK (Thomas and Palmer, 2014). However, this sample was not necessarily representative of the larger population of English primary teachers. Despite this limitation, the sample aimed to represent a range of teacher backgrounds, geographic locations and experiences which can be considered to represent the wider English year 6 teacher population. The findings related to teacher knowledge of place value and variable should also be transferrable beyond this sample since the literature identified the same problems in some teachers’ knowledge of place value and variable as this study.

The two case study teachers in Chapters 5 and 6 were representative of those teachers who had taken the ScratchMaths professional development and taught the first year of the curriculum. They can be considered as cases that had a high degree of fidelity (O’Donnell, 2008) for the ScratchMaths intervention as they participated in the intervention as initially designed. The third case study presented in Chapter 7 represents a teacher with a lower fidelity since they had experience of teaching Scratch but had attended no PD and no prior experience of teaching mathematics through programming. Combining the three cases with the data collected from the additional teacher episodes provided a more representative sample of teachers since these teachers who not taught Year 5 of the intervention, but had attended PD. The additional teacher episodes augmented the case studies to provide additional snapshots of teachers’ practices who were not observed repeatedly and thus reduce the Hawthorne effect which might be considered a limitation of the findings from the case study teachers.

I was the sole researcher of this study and have collected, analysed and presented all of the data. To minimise any potential researcher coding bias in my qualitative approach I adopted two methodological approaches. The first used Schoenfeld’s Teaching for Robust Understanding (TRU) framework the analysis of lesson observation. The framework had
developed over more than a decade of research of models of teaching and is grounded in the literature concerning mathematics teaching and learning. The framework was applied in a rigorous way by breaking up the lesson into a series of 10 minute episodes which were scored across each dimension and then calculated to provide a weighted average in each dimension for the lesson. The second methodological approach was the validation of both the TRU framework and my coded lesson observations by another member of the ScratchMaths research team who had experience in collecting and analysing qualitative data beyond the SM project. Our independent analysis of a single lesson produced mostly similar codes, and scores that were at most 0.5 apart. Together the methodological approaches support the reduction of researcher bias.

10.5 Implications for future research

There are several potentially productive lines of enquiry for future research. The first involves research into other fundamental mathematical topics from the SM curriculum such as geometric similarity or proportional reasoning. I realised during the SM intervention that we had created a wealth of curriculum material which remained untaught due to the impact of most teachers preparing for the Key Stage 2 mathematics tests after the first term of Year 6. I am proud to have collaborated with the project team on developing these materials and it is disappointing that the full curriculum was not able to be implemented and researched. Content within Module 5 of the SM curriculum focused on developing proportional relationships such as geometric similarity and content in Module 6 focused on transformations and coordinates, both key aspects of primary mathematics. This leads me to question whether these aspects of mathematics would also provide opportunities for teachers to confront their knowledge and (re)conceptualise it within programming.

The tension of teaching SM vs preparing for the KS2 tests was recognised by Mary from the case studies. Since finishing the intervention, Mary has addressed the tension by implemented the SM curriculum across a larger age range than initially designed to ensure that by the end of Year 6 all children have experience the full curriculum. Her approach has been to teach two modules per year from Year 4 to Year 6, and thus finish the curriculum before preparation
starts for the Year 6 assessments. This provides a second exciting research opportunity for the specific case of Mary’s school to examine the impact on children’s learning over the three year and how she has developed the teachers within the school to confidently teaching mathematics through programming.

The SM intervention was evaluated in 2018 by the Education Endowment Foundation (EEF) who found that engagement with SM had a positive impact on computational thinking but had no impact on mathematical attainment as measured by pupils’ performance in the Key Stage 2 test. This was disappointing although altogether not surprising given the low fidelity of curriculum implementation of SM within Year 6 (Noss et al., 2020). One criticism of the EEF evaluation is the use of the KS2 mathematics assessment at the end of the two year evaluation to measure children’s mathematical attainment across all of mathematics. The SM curriculum intention was to enable pupils to engage with and explore important key mathematical ideas through learning to program. The EEF evaluation had certainly provided robust evidence that exploring mathematics through programming does not impact children’s overall attainment in mathematics but raises the question of how their knowledge of key mathematical ideas has changed in ways beyond those measured by the KS2 assessment. Finally, the findings of this thesis have illustrated how teachers’ mathematical knowledge has evolved because of learning to program and professional development. Thus, a potentially rich area of research is to trace the impact of teacher professional development on children’s learning.
References


BECTA (2003). *What the research says about using ICT in maths*. Coventry: British Educational Communications and Technology Agency (Becta).


Bell, T. and Vahrenhold, J. (2018). 'CS unplugged—how is it used, and does it work?', *Adventures between lower bounds and higher altitudes* (pp. 497-521): Springer.


Implications for future research


Calder, N. S. (2018). 'Using Scratch to facilitate mathematical thinking'.


Close, G. S. (1982). 'Children's Understanding of Angle at the Primary/Secondary Transfer Stage'.


Department for Education. (2012). ‘Harmful’ ICT curriculum set to be dropped to make way for rigorous computer science
Implications for future research


Girit Yildiz, D. and Akyuz, D. (2019). 'Examining Two Middle School Mathematics Teachers’ Knowledge for Teaching Manipulation of Algebraic Expressions during Lesson Planning and Instruction'.


Workshop in Primary and Secondary Computing Education (pp. Article 8). Potsdam, Germany: Association for Computing Machinery.


Howe, R. (2019). 'Learning and using our base ten place value number system: theoretical perspectives and twenty-first century uses'. ZDM, 51 (1), 57-68.


Hoyles, C. and Noss, R. (2003). 'What can digital technologies take from and bring to research in mathematics education?'. In A. J. Bishop, M. A. Clements, C. Keitel, J. Kilpatrick
10.5 Implications for future research


Landry, G. A. (2010). 'Creating and validating an instrument to measure middle school mathematics teachers’ technological pedagogical content knowledge (TPACK)'.


Implications for future research


Miller, J. (2019). 'STEM education in the primary years to support mathematical thinking: using coding to identify mathematical structures and patterns'. ZDM, 51 (6), 915-927.


Noss, R. (1986). 'Constructing a conceptual framework for elementary algebra through Logo programming'. *Educational studies in mathematics*, 17 (4), 335-357.


Selling, S., Garcia, N. and Ball, D. (2016). 'What Does it Take to Develop Assessments of Mathematical Knowledge for Teaching?: Unpacking the Mathematical Work of Teaching'. [Online], 13, 35-51. Available at: [http://scholarworks.umt.edu/tme/vol13/iss1/4](http://scholarworks.umt.edu/tme/vol13/iss1/4).


Appendix A

A.1 Adapted Teaching for Robust Understanding (TRU) (Schoenfeld, 2014a; Schoenfeld, 2018) summary scoring rubric for analysing Teaching mathematics through programming

<table>
<thead>
<tr>
<th>Dimension 1</th>
<th>Dimension 2</th>
<th>Dimension 3</th>
<th>Dimension 4</th>
<th>Dimension 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics through programming</td>
<td>Cognitive Demand</td>
<td>Equitable access to content</td>
<td>Agency, Ownership and Identity</td>
<td>Formative assessment</td>
</tr>
<tr>
<td>To what extent does the teacher support exploration and development of mathematical ideas through learning to program?</td>
<td>To what extent are students supported in grappling with and making sense of mathematical concepts?</td>
<td>To what extent does the teacher support access to the content of the lesson for all students?</td>
<td>To what extent are students the source of ideas and discussion of them? How are student contributions framed?</td>
<td>To what extent is students’ mathematical thinking surfaced; to what extent does instruction build on student ideas when potentially valuable or address misunderstandings when they arise?</td>
</tr>
<tr>
<td>Basic (1)</td>
<td>The teacher makes limited or no use of technology, simply tells students what to do with no clear aim in mind. Student computer activity is unstructured, unfocused and with little or no use of the 5E’s.</td>
<td>Classroom activities are structured so that students mostly apply memorized procedures and/or work routine exercises.</td>
<td>There is differential access to or participation in the mathematical content, and no apparent efforts to address this issue.</td>
<td>The teacher initiates conversations. Students’ speech turns are short (one sentence or less), and constrained by what the teacher says or does.</td>
</tr>
<tr>
<td>Level</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proficient (2)</td>
<td>The teacher uses orchestrations and engages with the 5E pedagogical framework, but connections with mathematics (bridges) are limited. Students are given some, but limited opportunities to Explore, Explain, Envisage and Exchange when working with the computer. Classroom activities offer possibilities of conceptual richness or problem solving challenge, but teaching interactions tend to &quot;scaffold away&quot; the challenges, removing opportunities for productive struggle. There is uneven access or participation but the teacher makes some efforts to provide mathematical access to a wide range of students.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distinguished (3)</td>
<td>The teacher uses orchestrations and makes meaningful connections between representations in programming and representations in mathematics. (bridges). The teacher's hints or scaffolds support students in productive struggle in building understandings and engaging in mathematical practices. The teacher actively supports and to some degree achieves broad and meaningful mathematical participation; OR what appear to be established participation structures result in such engagement. Students have a chance to explain some of their thinking, but the teacher is the primary driver of conversations and arbiter of correctness. In class discussions, student ideas are not explored or built upon. The teacher refers to student thinking, perhaps even to common mistakes, but specific students' ideas are not built on (when potentially valuable) or used to address challenges (when problematic). Students explain their ideas and reasoning. The teacher may ascribe ownership for students' ideas in exposition, AND/OR students respond to and build on each other's ideas. The teacher solicits student thinking and subsequent instruction responds to those ideas, by building on productive beginnings or addressing emerging misunderstandings.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

B.1 An example of a lesson observation transcription

[Lesson starts at 13:00 of the video, after the pupils have collected a laptop in their pairs. Noticeably the script on the board is the script is ‘improved’ there is now a polygon block. The script doesn’t appear to have any bugs in it, although it’s not clear since the polygon definition block is off the screen.]

00:13:00
R: What did we do last lesson, what did we get to at the end? Talk to the person next to you, go!

00:13:11

[The pupils discuss for 20 seconds.]

00:13:30

R: Redina has an idea, what do you think?

P: I think we made a polygon block and a new variable.

R: Yes we did, we made a new variable! Can you remember what it was?

P: How many polygons?

R: Good. Zara, what else did we do?

P: We were trying to just the press the beetle, it would ask how many sides, what side length and how many polygons you want and then it would just do it!

R: What does she mean in Scratch language? Mustafa?

P: When the beetle sprite is clicked.

R: When the beetle sprite, when the sprite is clicked. Off you go, check that it does what Zara just described.

00:14:26

[The pupils check their work.]

R: Has yours worked anybody got any issues to solve, Shaima has!
S: When our sprite draws the polygon, there’s some at the bottom and some at the top?

R: What could Shaima do, there is some at the bottom and some at the top. What’s the first thing she could do?

M: Sort out the x and y axis.

R: That could be a solution, are we sure that we need a solution yet? How many times have you done it?

S: 2

R: She’s done it two times. What could we ask Shaima?

Z: Try it again.

R: Mustafa said to sort out the x and y axis, can you start thinking how you would tell her to sort it out. Have a look at your x and y axis that you put in there. Shaima has it done it again.

S: Yes.

R: Have you looked at your x and y coordinates. Can anyone help Shaima out? Khadiya?

K: Her y axis is not picked random.

R: Ahh, can you see it?

K: Yes.

[R points to the board to the go to x pick random, from -240 to 240.]

R: Go to x pick random, from: 240, we talked about it being right the way over there. [she points to the far left of the screen], to 240 right at this edge, and we looked at y pick random,
we didn’t have it all the way to the bottom, we found out where it stopped [she points] minus 85, up to the top. Does your say that? Good problem solving!

00:17:10

R: I’ve got a problem with mine. If I click my beetle right now, what will happen? Talk to the person next to you go through it step by step.

00:17:20

[Pupils discuss for about 20 seconds]

00:17:53

R: What will happen? Redina very cleverly looked at her script and compared it with mine and she spotted something that was difference. But can you explain what that difference will do?
What will happen on my screen? How does that difference in my screen, what does it look like?

P: The repeat at the bottom, you’ve got polygon, set random pen shade and goto, but in our one we’ve got a pen up and pen down.

R: You’ve told me the difference, but that’s not what I’m asking. You’ve told me there’s a difference in the code, but I want to know how it will LOOK different when I run the script, when I click the sprite and I run the script. What will happen that’s different? Jack?

J: When you click the sprite I think it will go everywhere but not make any polygons and draw them in the sky.

R: Does anybody agree or disagree? What do you think Drishti?

P: I agree with Jack, I think it will just go to a random position but won’t leave anything behind.

R: So it won’t do any drawing, why do you think that Jack?

J: Because there’s no pen up or pen down

R: But there is a pen down somewhere in my script?

R: Who disagrees? Mahi

M: I disagree with Jack and Drishti because in our set up script, it has pen down already, so we won’t need another pen down.

R: Can you see that Jack? So Jack and Drishti had an idea, we’ve decided they were wrong, who has another idea? Iftin?

I: I think you need a pen up?
R: Where?

I: After the set random pen shade

R: But why, what would it look like if I ran the script now, you’ve solved the problem for me, but you haven’t answered my very first question which was what will it look like, what will happen when I run the script. Demi?

D: It will go to numbers but it won’t draw.

R: We had that guess but it wasn’t quite right because we do have a pen down [she points to it]. Marion?

M: All the shapes will be drawn together with lines

R: Why?

M: [Unintelligible]

R: Marion is saying it will draw my polygon, and when it moves it will still be drawing, so it will draw the line as it moves [she gestures this]. How will I solve that problem? Talk to the person next to you, quickly.

00:20:38

00:20:50
R: Right Elton, what do I do?

E: You need a pen up, before Polygon.

R: Where should I put it?

E: Before Polygon

R: Do we agree, before Polygon?

Pupil in unison: Nooooo

R: Anil what do you think?

A: I think before Polygon, but…
R: Ok stop, if I do pen up before Polygon. What is the role of Polygon, Mustafa?

M: It draws a polygon.

R: If I do pen up before it draws a polygon, what will happen Elton? [She gestures a pen up with her hand.]

E: Ohh, after!

R: What will happen though, if I do pen up before?

E: It won’t draw anything.

R: It won’t draw anything! So where does it need to go Elton?

E: After polygon.

R: And before what? What does pen up need to go before. We need to lift it up before something happens?

E: Set random penshade, no go to x.

R: Why, Marion, why is it before go to x.

M: Because if it does it after goes to x it’s going to start moving.

R: Do we want to draw a line as it goes somewhere? [She gestures the movement.]

Pupils in unison: No!

R: So we need pen up there, so fixed it then. There we go fixed!

[She moves the pen up to before the go to.]
P: No!

R: No, Anila?

A: I think we need a pen down after the go to.

[She drags a pen down after the go to.]

R: Pen down, after go to, anywhere else where you could put it?

P: Under the repeat block.

R: Under the repeat block, so down here. What do you think? Who’s got an opinion. Drishti?

D: Do it before polygon.

R: We could it before Polygon, would it work if we put it there.

[She puts the block outside of the repeat at the bottom of the script.]
R: What do you think, Rohan?

P: Oh.

R: He’s not sure. Farzana?

F: I think it won’t work because the last thing, is how many polygons, it will do that many polygons, and then after it has done that it will be pen down.

R: N start thinking please, we don’t want it outside of the repeat, do we. We then said we could put it there. [She moves it to under the goto x]. Any other positions in the script, Iftin?

I: Before the polygon?

R: Will that work?
Pupils: Yeah!

R: Let’s try, a little triangle today, 8 and 25. Does it work?

[She demonstrates the script with 3, 8, 25]

Pupils: Yeah

00:23:40

R: Today we are going to take our script a little further by the end of today!

[She shows the following, the children in unison woop and wow!]
R: Which bits of this can we already do, which bits have we got sorted? Naseema?

N: …

R: Jamie, whisper to Nazeema!

P: The fireworks!

R: The fireworks, we can make this happen. Which bit haven’t we done yet? Which bit haven’t we done yet? Yousef?

Y: The bar models

P: Ladders!

R: Nice! Not bar models not in our landscape. Shaima?
S: I think they are an outline of a building.

R: They are, what word could we call them?

P: Skyscrapers.

R: Skyscrapers, we talked about them the other day. Today we are going to be draw some skyscrapers, so keep that picture in mind.

R: Today we are going to continue with the project that you’ve got. I’m going to just give this to you because you know so much about Scratch, you can do this. You are going to make a new block, call it square, what do you think you are going to make the block do. What is it going to do?

P: Square,
P: A random square.

R: A random square anywhere? [Shakes her head]

P: A square.

R: What code needs to into the code to make your block, and then you are going to use a side length variable to draw it. So you are going to get it to ask, what’s the side length. Does it need to ask anything else? Zara?

Z: The side length

R: We need to ask how long the side length is. What other questions you could ask, look at the questions you have you been asking, they are on your screen. I want to draw one square, Farzana.

F: Um.

R: Look at the questions you have asked so far, Shaima?

S: How many sides?

R: How many sides, do we need to ask how many sides, is that going to change, because?

P: It’s a square.

R: It’s a square, I’ll leave you there.

00:26:45

00:32:00
R: I will show you what some of you are doing and want you tell me whether this is right and what I could change.

R: So I’ve defined a new block called square, I’ve seen lots of people doing very good things, they’ve got a repeat, they’ve got their move, they’ve got their turn, and I’ve seen lots of people knowing that the degree of turn for the square will be?

Pupils: 90!

R: Everybody got that, so I’m very impressed, and they put that in their repeat, and they’ve looked at what they were asked, [she switches back to the ppt slide] and they were asked to use the side length variable, so they’ve gone and got their side length variable and they’ve done this.

P: Yay!

R: What do you think, talk to the person next to you!

00:32:59

[The pupils burst into discussion for about 10 seconds]
R: What is the problem, Mahi?

M: What the problem of the script is, if it’s repeating by the side length, then it won’t work because…

R: Ok pause, can you tell me which bit of my script here is the side length? Iftin?

I: The move.

R: What do you think Mahi?

M: The move.

R: So where should my side length variable go, Redina?

[She moves the side length variable out of the repeat.]

R: The repeat?

R: It was in the repeat but we decided that wasn’t the right place for it. So where would it?

Does the repeat do the side length? Jack?

J: I think the side length goes into the move?

[She moves the side length variable into the move block]

R: Do we agree? If we have it in the move, what’s going to go in the repeat then?

RC: What’s going to go in the repeat, if we put the side length in the move.

00:34:30
R: What’s going to go in the repeat, Malaika?

M: I think it’s going to be 4.

R: Why?

M: Because 90 times 4 is 360

R: Which is a maths fact, but also 4 add 5 is also 9.

M: Because 360 degrees is a full turn.

R: OK, can anyone help her? She knows it’s got 360 degrees in a turn. What does the repeat actually do when we are drawing, Drishti?

D: It repeats the move and the turn, if we didn’t have the repeat, it won’t, it would just draw one line, we would have to repeat it to make the square.

R: Super, because we want 4 sides, don’t we? [She gestures the 4 sides moving with her hand].

R: When I click my sprite, will it draw a square for me?

P: Yes!

P: No!

R: Some say Yes, some say No. What do you think? Tell the person next to you.

00:35:31

[…]

00:35:44
R: Khadiuat, when I click the sprite, will it draw a square?

K: When you click the sprite, I think it will draw a square.

R: Ok let’s do it.

K: I don’t think so, because you haven’t said what the side length is, so whenever they ask you the question?

R: Can you see my screen, or do you need to move to the carpet? Look where it says when this sprite is clicked, Iftin?

I: I think you need to click green flag because it has pen down.

R: Green flag, got pen down, so I click it, how many sides! wait a minute, I’ve told it how many sides, it’s a square. Why is it asking me how many sides? Mustafa?
M: Because you already have one block that says when this sprite is clicked, so it’s referring to that. R: Ahh can you see that this is lit up here, does this draw a square?

P: No.

R: So if we separate this, what do I need to add to when this sprite is clicked? Shall I stick this one onto it?

P: No.

R: Oh it doesn’t work.

P: You can’t.

R: Ahh you can’t, off you go and do it please.
00:37:04

[Pupils work on the task.]

00:38:43

R: If you have done that, can you please think about the question that is on the board, if you have one square drawn, start thinking about that question. Remember we are drawing skyscrapers, we are going to have a square, a square, a square, each of those squares will be a new floor!

00:39:43

R: Turn yourselves around. Who has some ideas about that question, let’s think about what you are looking to draw, is a set of squares, one on top of the other.
R: After I’ve drawn one square, I’m going to move to draw the next square. Anyone got any ideas, look at what I just did, what is my pen doing after each square? [She draws out a second tower of squares.] Bradley?
B: Is it moving up?

R: It’s moving up!

00:40:44

R: I wanted to ask you something before we do this. If we look at my square, does it matter if I have my move and turn the other way around? This one it moves and then turns, the next one it turns and then it moves. Get out your white board and a pen please and follow both of those scripts through.

00:41:23

[The pupils work on this task until 00:43:40]

00:43:40

R: Pens down, keep that idea in your head.

R: If I keep this one, for my definition of a square. To draw the tower, someone said we are going to draw a square and then move, and then we are going to do another square, and then another move. Have a go when you do that, off you go.
00:44:19

[Pupils work on the task until 00:51:11.]

00:51:11

R: Right, a lot of you are at the point when you can draw one tower. Can anyone not do that yet?

_A couple of hands go up._

R: I will come around and help you.

R: I want you to build a script, that asks the side length. A lot of you have put what’s the side length into your definition of the square, can you take it out of there and put it into this script.

_[R notices a mistake in the slide, pauses for a moment, that there is a missing square block._]

R: You need to draw a tower that is 10 squares tall, off you go!

00:52:11
[The pupils work until 00:55:45]

00:55:45

R: I’ve noticed a lot of people’s problems, some of you have this, and some of you don’t. When you draw a square, you need to find out where does your sprite stop. Mine just stopped there.
R: If I wanted to draw my next square on top of it, how far does it need to move?

P: The same.

R: The same as what?

P: The same as the side length!

R: Yes, the same as side length, I’m going to use some algebra. M for Marion. My side length is M that’s my variable. In yours it’s not called Marion it’s called side length. Sorry Marion you’re out. So, I’ve moved the side length, how far do I move in this direction?

Pupils in unison: Side length.

R: How far do I move in this direction?

Pupils in unison: Side length.

R: How far do I move in this direction?

Pupils in unison: Side length.

R: Because in your block that says move, you are not moving a number you are moving a?
Pupils in unison: Side length.

R: Every time you go around. So if I stop here, N just said I need to move the same. So what do I need to move?

[Image of a drawing of a square with a circle inside and a note written on it]

Pupils in unison: Side length.

R: So you need to use that side length variable in another place in your script, so make sure that every time it draws, it does it. So check to see if that’s your problem, have a go.

00:57:41

[Pupils work on the task until 01:02:11]

01:02:11

R: Anyone except for Jamie and Brdyl who hasn’t made a tower. Shaima and Seema, let’s see if we can solve Shaima and Seema’s problem.
R: EVERYBODY turn your chair to see the board.

*[R opens up their work from the Scratch interface, several pupils move to the carpet]*

R: What should we check works first? Malaika?

P: The square block.

R: Let’s check does the square work?

*[She presses green flag, clicks the square block, and it draws a square]*.

R: Does the square work?

Pupils in unison: Yes!

R: Then we have when this sprite is clicked, it’s going to do something 10 times, it’s going to ask what’s the side length, set the side length to the answer, draw a square, move 20 steps and turn 90 degrees.

B: That’s a lot.

R: I was just thinking that. Shaima and Seema what don’t we need to do 20 times.

S: Ask the question.
R: So where could we put the question? I’ve taken them out of the repeat. Where could the question goes.

S: Into the define square.

R: If we put it in define square, it would still be in this repeat, it would still be repeated 10 times. Mustafa?

M: You can put it on top of the repeat.

R: Can you see that if we put it out of the repeat so it won’t happen ten times?

R: Shaima can you see that.

R: Then in here, we are going to draw the square, we are going to move 20 and turn 90 degrees. Seema and Shaima, what you are saying is this.

01:04:54

R: Click me!
Rosie plays beetle.

R: How big do you want the square to be, Khadiyat?

K: 10 side length?

R: Off I go, 10, 10, 10, 10, move 20 turn 90, 10, 10, 10, 10, move 20, turn 90, is that what you wanted?

Pupils: No

R: No. What’s the issue, what’s the problem? Demi?

D: I think you should move 20 [unintelligible]

R: Well for this shape my side length was 10, and my move was 20. So 10 here, move 20, is what they wanted?

Pupils: No
R: What if my side length was 5, you move, you draw 5, and then you move 20. Is that what we want?

Pupils: No

R: Kahdiyat, what’s the problem.

K: You need to move by side length steps, so that the sprite always moves by the side length that you have told it.

R: Can you see that Shaima? This is the side length, it’s that variable called the side length, isn’t it? We need to make sure it jumps up to the top of the side length to draw the next one. Do I want move 20 steps?

R: I would want move how many steps?

D: 10

R: 10 Drishti says! Will it always be 10?

Pupils in unison: Side length.

R: if I find my variable side length and try it again now.

01:06:54

[R tries the script after moving in the side length variable she tries 50]
R: Oh what’s happened now? Mahi?

[A pupil calls out you need a move.]

M: You don’t really need a turn, because it has already turned.

R: Have a look at what happens, which direction is it pointing in when I do square?

M: Pointing up.

R: It’s pointing upwards, so do I need to turn it Shaima? No, I can get rid of my turn, shall we try it now.

[R tries the script]
R: Have we solved Shaima’s problem?

Pupils: YES!

[Pupils clap.]

End of lesson

01:07:42

Note, Shaima’s script actually does the following, because of the turn which Rosie had ignored when she acted out and played Beetle.
### Appendix C

#### C.1 An extract from a lesson observation spreadsheet

<table>
<thead>
<tr>
<th>Episode</th>
<th>Start</th>
<th>Length</th>
<th>Description</th>
<th>Example quote</th>
<th>Explain</th>
<th>Explore</th>
<th>Envelope</th>
<th>Exchange</th>
<th>Bridge</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Episode 1</td>
<td>00:13:47</td>
<td>00:01:08</td>
<td>Discussion question</td>
<td>What were we doing, reminded me where we were up to, children?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reminder of the previous lesson. The teacher wanted the children to use the phase polygon.</td>
</tr>
<tr>
<td>Episode 2</td>
<td>00:18:50</td>
<td>00:00:19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 3</td>
<td>00:15:09</td>
<td>00:01:22</td>
<td>Direct question</td>
<td>Ralph, what were we doing?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 4</td>
<td>00:20:31</td>
<td>00:01:19</td>
<td>Scratch-discuss the screen</td>
<td>Is this a triangle? Which part of the shape have we got? What will repeat 25?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Repeat 25 is the number of children's turns in the polygon definition, e.g., the number of sides or sides.</td>
</tr>
<tr>
<td>Episode 5</td>
<td>00:23:50</td>
<td>00:00:47</td>
<td>Scratch-discuss the screen</td>
<td>So this shape, the repeat is the answer. Why is the repeat the answer?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 6</td>
<td>00:22:37</td>
<td>00:00:53</td>
<td>Scratch-discuss the screen</td>
<td>Turn 360 degrees by answer, what on earth is that all about?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>To calculate the angle of turn for a shape, 360/answer. Also asks what is 90/4.</td>
</tr>
<tr>
<td>Episode 7</td>
<td>00:22:30</td>
<td>00:00:53</td>
<td>Scratch-discuss the screen</td>
<td>So then I've got these bits here, the points to the green block, pick up random. It's going to draw the shape and then go to somewhere else. Where do we want it to go to? So my sprite is going to draw a shape, shall we see what it does at the moment?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 8</td>
<td>00:24:23</td>
<td>00:00:09</td>
<td>Scratch-discuss the screen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 9</td>
<td>00:24:32</td>
<td>00:00:53</td>
<td>Scratch-discuss the screen</td>
<td>Is that what we wanted it to do? Pupil-down. Well, pen-down is in my set up script, and I've pressed my green flag. So pen-down, we're let's do it.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A pupil says pen down, but there's points to this in the setup script, so she needs that. A shape is drawn but it goes around it over and over in one place. A pupil says they need random position. The blocks are added but the default values of 0 to 0. The script is fine, but no change to the output.</td>
</tr>
<tr>
<td>Episode 10</td>
<td>00:24:32</td>
<td>00:00:53</td>
<td>Scratch-discuss the screen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 11</td>
<td>00:25:24</td>
<td>00:00:09</td>
<td>Scratch-discuss the screen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episode 12</td>
<td>00:25:30</td>
<td>00:00:37</td>
<td>Scratch-discuss the screen</td>
<td>Do we want it to go anywhere, do we want to have line ends on the ground?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A pupil notices that 0, 0 is the middle of the screen, then 0, 0 and 0, 0.</td>
</tr>
</tbody>
</table>
Appendix D

D.1 An example of an interview transcription

Question 1
It looks to me like they are just counting the digits, and seeing that 31.3 has 3 digits, and then looking that, the 3 in the 31 is a hundred so they are just discounting the decimal, have no understanding of the decimal at all.

Question 2
This is interesting, we don't do it like this.

PS - How do you do it?

Underneath, I'm assuming that's what this is. These two A and B, it appears they are carrying over ten instead of twenty, but then this one, this one is different, or is it, they've put that the first column adds up to 14 which it doesn’t, it adds up to 15, does it? My year two brain on a Friday afternoon, honestly you forget it straight away. A and B have the same kind of error and the error is that, when they've done it in the past with two digit numbers, they will have known that it's usually a ten that they carry over, but because it’s the third [three two] digit [numbers], they've carried over just a ten again, which I assume that is, just a ten again, when it's not a ten. Is that right?

PS - Which has a better understanding of this algorithm?

Probably C, because their error is just with a simple calculation at the beginning although it's probably hard to say, because the answer isn't a twenty number, so I can't really, maybe you could give them one that is the same.

Question 3
OK this is a really simple place value issue; she is seeing the 2 as two and not 20. I see that all the time, I tend to not use the individual counters, but tend to use the Diene, the base 10, because otherwise they do.

PS - Which language do you use, carry or exchange?

SW - Kind of both since either can appear in the test, so we tend to switch between the two.

At the end we might have a more formal, towards the end. We tend to represent it more as a picture, they draw the sticks in a column format, so the 45 and the 47. The higher achievers would do the three. And they would physically take then ten and swap it and things like that to help them to understand, so it's an introduction into this [the formal algorithm]

Question 4
Two individual sprites working at the same time, it will be set so that when the ones sprite is clicked it changes to the next costume which is the next number. When it gets to ten, it will appear as a 0 but it will broadcast a message at the same time so that the next sprite changes to the next costume and that keeps going.
PS - Why is it ten, why is costume# =10?

Laughs, I know that's that the case, because err, it has become ten, it's after nine, they always ask me that, and I'm like err. They know it as well, yeah... [sounds unsure].

**Question 5**

I would get rid of scripts, cos the scripts would be duplicated. I would change all the set up scripts first, then make sure that the tens sprite is set up to broadcast the message to the hundreds and then the hundreds receives the message, so you would probably edit the script rather than building a new one.

PS - How would you test it?

Mainly set the digits so that it's an 8, maybe the tens is an 8, so that they are not clicking for ages. Green flag. Get them to try it a few times, make sure that it works.

**Question 6**

Too big for Year 2, so thinking of Year 5. I would probably, represent it as drawing it, and then looking at the effect on the units and the tens. They would probably use their number bond knowledge. So, the children will probably say straight away, or this what we teach them to, they know that nine add one is ten, so then that straight away will change to 90 and then they have three more, so they use the number bonds. That's more likely what we would do.

PS - In scratch?

I wish I could see it, I'm really visual. By clicking 4 times, that would work [not change it so it adds on 4]. And if it were add 4, I could change the green, so that it's next costume add 4.

Probably with each click, when I click, I ask them what is going to happen and why ,so they do some predicting, because I think that's what I try to do, so they show me that they are understand it before they do it, rather than do it and explain afterwards. Especially from the 9 that first step, as they understand it from 1 upwards.

**Question 7**

The ones stays the same, it's the ten seconds, once that gets to 59, the next time, when it's a 6 is when it is different. That's the first time we've done that, so I'm still trying to get my head around it myself.

PS - How is this similar to the place value model?
In Scratch or in general?

PS - You said today "what's the same and what's different?". The numbers are kind of reacting to one another, but that's not what I mean. So ask me one more time?

PS - How is the time model related to the place value model previously.

So when the one gets to a certain point, that has to make the tens change and that happens all the way through for the four sprites. But obviously they change at different positions... Sorry.

**Question 8**

I don't make it obvious to them, when we first started with the flip cards, I didn't tell them that we were focusing on place value, we just played the game and looked at it without explaining, and then related it back during the lesson rather than straight away. I try and to that quite a lot, so I can see, use that as an assessment to see what their understanding is of the topic, that was why I did that a little bit today with time, have a little look at it, what do you think is going to happen, and then relate it back and talk about it during it, rather than at the beginning.

**Question 9**

Not year 6 teacher, difficult to answer.

**Question 10**

I think it's more, explaining the processes has been the main thing, it's really influenced quite a lot thinking about place value, cos I think it was always one of those things, where I knew it was the case and explained it, but actually now, with more of my Scratch head on, this happens and then this makes this change and that kind of thing. I explained it more to the Year 2's, using that kind of language. I actually showed it to them, I didn't explain how it works, I showed them the model so that they could actually another way of physically changing the numbers. But I think that generally even with a lot of the thing things last year, it's been most helpful when teaching problem solving and getting the children to break down the problem in the way that it would break down the scripts to understand them and take them apart and stuff, so yeah.
Appendix E

E.1 An example of a TRU calculation

The table shows the five phases of the lesson broken up into approximately 10 minute sections. Each phased was scored across the five dimensions and given a score from 0 to 3, using an interpolated 5 point scale using the rubric shown in Appendix A. The weighted average is then calculated for each dimension per phase.

E.g., weighted average for Dimension 1.

\[
\frac{(3 \times 640 + 2.5 \times 804 + 2 \times 847 + 3 \times 823 + 2 \times 168)}{3282} = 2.6
\]

<table>
<thead>
<tr>
<th>Phase</th>
<th>Episode length mins and seconds</th>
<th>Episode length in seconds (*86400)</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>00:10:40</td>
<td>640.00</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Phase 2</td>
<td>00:13:24</td>
<td>804.00</td>
<td>2.5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>Phase 3</td>
<td>00:14:07</td>
<td>847.00</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Phase 4</td>
<td>00:13:43</td>
<td>823.00</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Phase 5</td>
<td>00:02:48</td>
<td>168.00</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Weighted average

| Weighted average | 0:54:42 | 3282.00 | 2.6 | 2.7 | 3.0 | 2.3 | 2.2 |
Appendix F

F.1 Ethics Application Form: Student Research

All research activity conducted under the auspices of the Institute by staff, students or visitors, where the research involves human participants or the use of data collected from human participants are required to gain ethical approval before starting. *This includes preliminary and pilot studies.* Please answer all relevant questions responses in terms that can be understood by a lay person and note your form may be returned if incomplete.

For further support and guidance please see accompanying guidelines and the Ethics Review Procedures for Student Research [http://www.ioe.ac.uk/studentethics](http://www.ioe.ac.uk/studentethics/) or contact your supervisor or researchethics@ioe.ac.uk.

Before completing this form you will need to discuss your proposal fully with your supervisor(s).
Please attach all supporting documents and letters.

*For all Psychology students, this form should be completed with reference to the British Psychological Society (BPS) Code of Human Research Ethics and Code of Ethics and Conduct.*

<table>
<thead>
<tr>
<th>Section 1 Project details</th>
<th>Learning to Scratch: the evolution of teachers’ mathematical knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Project title</td>
<td></td>
</tr>
<tr>
<td>b. Student name and ID number (e.g. ABC12345678)</td>
<td>Piers Saunders</td>
</tr>
<tr>
<td>c. Supervisor/Personal Tutor</td>
<td>Richard Noss</td>
</tr>
<tr>
<td>d. Department</td>
<td>CCM</td>
</tr>
</tbody>
</table>
### Course category (Tick one)

<table>
<thead>
<tr>
<th>Course Category</th>
<th>PhD/MPhil</th>
<th>EdD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTeach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diploma (state which)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other (state which)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Course/module title

- **MPhil/PhD in Education**

### If applicable, state who the funder is and if funding has been confirmed.

### Intended research start date

- May 25th 2016

### Intended research end date

- Dec 30th 2018

### Country fieldwork will be conducted in

- **UK**

### Has this project been considered by another (external) Research Ethics Committee?

- Yes  
  - External Committee Name:

- No
  - **go to Section 2**
  - Date of Approval:

**If yes:**
- Submit a copy of the approval letter with this application.
- Proceed to Section 10 Attachments.
Note: Ensure that you check the guidelines carefully as research with some participants will require ethical approval from a different ethics committee such as the National Research Ethics Service (NRES) or Social Care Research Ethics Committee (SCREC). In addition, if your research is based in another institution then you may be required to apply to their research ethics committee.

Section 2 Project summary

Research methods (tick all that apply)

*Please attach questionnaires, visual methods and schedules for interviews (even in draft form).*

- Interviews
- Focus groups
- Questionnaires
- Action research
- Observation
- Literature review
- Controlled trial/other intervention study
- Use of personal records
- Systematic review ⇒ *if only method used go to Section 5.*
- Secondary data analysis ⇒ *if secondary analysis used go to Section 6.*
- Advisory/consultation/collaborative groups
- Other, give details:

Please provide an overview of your research. This should include some or all of the following: purpose of the research, aims, main research questions, research design, participants, sampling, your method of data collection (e.g., observations, interviews, questionnaires, etc.) and kind of questions that will be asked, reporting and dissemination (typically 300-500 words).

Background and Design

This study intends to show that through learning to program in Scratch teachers can be helped to build an epistemological map of connections between mathematics and the programming instantiation of the mathematical idea which shapes their mathematical knowledge for teaching. The study will be situated within the context of the ScratchMaths project, an initiative to design materials for year 5 and year 6 teachers to address the demands of the new computing curriculum in the UK and to develop computational and
mathematical thinking skills. The materials include the design and delivery of a two day continued professional development (CPD) course.

**Participants and sampling**
Approximately 100 primary schools have volunteered to be involved in the project. Head teachers of the schools involved have signed a Memorandum of Understanding with the project agreeing to take part in the project work. Copy attached. 50 schools selected through a randomized controlled trial (RCT) managed by Sheffield Hallam University (SHU) will take part in the CPD.

The main data for the study will be collected through a longitudinal study of 8 teachers defined by the following categories i) computing/coding teachers, ii) KS2 maths teachers (year 5 or 6), iii) experienced teacher across all key stages and iv) new teachers to the profession.

Additional data will be collected from teachers during the CPD as outlined below.

**Methods**
Data collection and ongoing analysis will begin in May 2016 and end in December 2017. At minimum there will be 5 data collection points during the data collection phases over the 18 months:

**Phase 1 – all teachers attending CPD**
- before the CPD (May 2016),
- during the CPD (May and June 2016)
- at the end of the two days of CPD (June 2016),

**Phase 2 – 8 identified as above**
- several times during the 2016-2017 school year following teaching a module,
- at the end of teaching all of the modules, post KS2 SAT – June 2017

**Phase 1 -** A preliminary questionnaire is being sent out to all CPD participants (attached), the kinds of questions asked are to collect participant’s previous experience of using SCRATCH. Following the initial data collection, the teachers scheduled to attend the CPD will be sent a computer based Scratch task. The task will be constructed to provide teachers an opportunity to engage with a mathematical idea whilst coding. The artefacts will provide an initial insight into Scratch confidence and competence and become an object to use within the task based interviews developed for phase 2 and during the CPD itself.
During the CPD, teachers will engage with pupil scenario tasks which were effective at providing a window on teacher thinking and reasoning in pilot testing last year. The authentic tasks are based on situations from the classroom and encourage teachers to identify pupil approaches, possible errors to propose what they would do next to move the pupil forward. Working with a small group of 4 teachers selected from each CPD session, responses will be audio captured and notes taken.

During phase 2, the primary data will be collected during multiple semi-structured task based interviews throughout the teaching year following the teaching of a module of ScratchMaths, e.g. following the Building with Number module. The focus of interviews will be to explore potential restructurations of teacher’s mathematical knowledge. An instrument as yet undeveloped will be based upon Selling, Garcia and Ball (2016) framework for assessing mathematical knowledge for teaching at the elementary level.

During phase 2, 3 teachers will be video and audio recorded as they teach Scratch Maths. The video will be focused upon the teacher only. All three schools have signed consent and contracts in place with pupils/parents that video recording can take place in the classroom should any pupils be captured in the video/audio.

Section 3  Participants

Please answer the following questions giving full details where necessary. Text boxes will expand for your responses.

a. Will your research involve human participants?  Yes ☒  No ☐ go to Section 4

b. Who are the participants (i.e. what sorts of people will be involved)?  Tick all that apply.
   - Primary school teachers
   - Unknown – specify below
   - Adults please specify below
   - Other – specify below

NB: Ensure that you check the guidelines (Section 1) carefully as research with some participants will require ethical approval from a different ethics committee such as the
National Research Ethics Service (NRES).
Primary school teachers

c. If participants are under the responsibility of others (such as parents, teachers or medical staff) how do you intend to obtain permission to approach the participants to take part in the study?

(Please attach approach letters or details of permission procedures – see Section 9 Attachments.)

d. How will participants be recruited (identified and approached)?
Participants are recruited as a result of the parent project ScratchMaths which has received ethical approval.

e. Describe the process you will use to inform participants about what you are doing.
During the briefing session at start of CPD, the project team will be introduced and participants informed that case notes will be taken as a result of observation and participants may be asked questions related to their previous experiences with SCRATCH or related to their activity within the CPD and audio recorded. Before the lesson observation, participants will be informed that video and audio recording of the lesson will take place. The participants have the option of withdrawing at any time from being recorded.

f. How will you obtain the consent of participants? Will this be written? How will it be made clear to participants that they may withdraw consent to participate at any time?

See the guidelines for information on opt-in and opt-out procedures. Please note that the method of consent should be appropriate to the research and fully explained.

Participants have signed a MOU for the parent ScratchMatch project – see attached – extract from MOU “Allow participating staff to take part in research interviews, surveys, events and videos as required by the project within reasonable scope of their time and availability.”

g. Studies involving questionnaires: Will participants be given the option of omitting questions they do not wish to answer?

Yes ☒ No ☐
If **NO** please explain why below and ensure that you cover any ethical issues arising from this in section 8.

h. **Studies involving observation**: Confirm whether participants will be asked for their informed consent to be observed.
   - Yes ☒ No ☐
   
   If **NO** read the guidelines (Ethical Issues section) and explain why below and ensure that you cover any ethical issues arising from this in section 8.

i. Might participants experience anxiety, discomfort or embarrassment as a result of your study?
   - Yes ☐ No ☒
   
   If **yes** what steps will you take to explain and minimise this?
   
   **If not**, explain how you can be sure that no discomfort or embarrassment will arise?
   
   For the purposes of the pilot background data gathering, I will take observation notes of teachers during the CPD sessions acting as a participant observer. Participants will be informed of this before the CPD session following on from their agreement within the MOU. The unstructured interviews will focus on participant’s experiences of the CPD and previous experience of using SCRATCH programming avoiding emotional responses. The participants have agreed to be video recorded, and have the option of withdrawing their consent at any time.

j. Will your project involve deliberately misleading participants (deception) in any way?
   - Yes ☐ No ☒
   
   If **YES** please provide further details below and ensure that you cover any ethical issues arising from this in section 8.

k. Will you debrief participants at the end of their participation (i.e. give them a brief explanation of the study)?
   - Yes ☒ No ☐
If **NO** please explain why below and ensure that you cover any ethical issues arising from this in section 8.

<table>
<thead>
<tr>
<th>l.</th>
<th>Will participants be given information about the findings of your study? (This could be a brief summary of your findings in general; it is not the same as an individual debriefing.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>☐ No ☐</td>
</tr>
</tbody>
</table>

**If no, why not?**

### Section 4 Security-sensitive material
*Only complete if applicable*

Security sensitive research includes: commissioned by the military; commissioned under an EU security call; involves the acquisition of security clearances; concerns terrorist or extreme groups.

<table>
<thead>
<tr>
<th>a.</th>
<th>Will your project consider or encounter security-sensitive material?</th>
<th>Yes ☐ * No ☐</th>
</tr>
</thead>
<tbody>
<tr>
<td>b.</td>
<td>Will you be visiting websites associated with extreme or terrorist organisations?</td>
<td>Yes ☐ * No ☐</td>
</tr>
<tr>
<td>c.</td>
<td>Will you be storing or transmitting any materials that could be interpreted as promoting or endorsing terrorist acts?</td>
<td>Yes ☐ * No ☐</td>
</tr>
</tbody>
</table>

*Give further details in Section 8 Ethical Issues*

### Section 5 Systematic review of research
*Only complete if applicable*

<table>
<thead>
<tr>
<th>a.</th>
<th>Will you be collecting any new data from participants?</th>
<th>Yes ☐ * No ☐</th>
</tr>
</thead>
<tbody>
<tr>
<td>b.</td>
<td>Will you be analysing any secondary data?</td>
<td>Yes ☐ * No ☐</td>
</tr>
</tbody>
</table>
* Give further details in Section 8 Ethical Issues

If your methods do not involve engagement with participants (e.g. systematic review, literature review) and if you have answered No to both questions, please go to Section 10 Attachments.

### Section 6 Secondary data analysis  Complete for all secondary analysis

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>Name of dataset/s</td>
</tr>
<tr>
<td>b.</td>
<td>Owner of dataset/s</td>
</tr>
</tbody>
</table>
| c. | Are the data in the public domain?  
   | Yes ☐  No ☐  
   | **If no, do you have the owner’s permission/license?**  
   | Yes ☐  No* ☐ |
| d. | Are the data anonymised?  
   | Yes ☐  No ☐  
   | **Do you plan to anonymise the data?**  
   | Yes ☐  No* ☐  
   | **Do you plan to use individual level data?**  
   | Yes* ☐  No ☐  
   | **Will you be linking data to individuals?**  
   | Yes* ☐  No ☐ |
| e. | Are the data sensitive ([DPA 1998 definition](#))?  
   | Yes* ☐  No ☐ |
| f. | Will you be conducting analysis within the remit it was originally collected for?  
   | Yes ☐  No* ☐ |
| g. | If no, was consent gained from participants for subsequent/future analysis?  
   | Yes ☐  No* ☐ |
| h. | If no, was data collected prior to ethics approval process?  
   | Yes ☐  No* ☐ |

* Give further details in Section 8 Ethical Issues

If secondary analysis is only method used and no answers with asterisks are ticked, go to Section 9 Attachments.
## Section 7 Data Storage and Security

*Please ensure that you include all hard and electronic data when completing this section.*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>Confirm that all personal data will be stored and processed in compliance with the Data Protection Act 1998 (DPA 1998). <em>(See the Guidelines and the Institute’s Data Protection &amp; Records Management Policy for more detail.)</em></td>
</tr>
<tr>
<td>b.</td>
<td>Will personal data be processed or be sent outside the European Economic Area?</td>
</tr>
<tr>
<td><em>If yes, please confirm that there are adequate levels of protections in compliance with the DPA 1998 and state what these arrangements are below.</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Who will have access to the data and personal information, including advisory/consultation groups and during transcription?</td>
</tr>
<tr>
<td>c.</td>
<td>During the research</td>
</tr>
<tr>
<td>d.</td>
<td>Where will the data be stored? Encrypted hard drive IOE/UCL laptop and written field note journal kept under direct supervision of myself, or locked in office at UCL. Questionnaire data on UCL network (not cloud).</td>
</tr>
<tr>
<td>e.</td>
<td>Will mobile devices such as USB storage and laptops be used? <em>No</em></td>
</tr>
<tr>
<td><em>If yes, state what mobile devices: IOE/UCL Laptop</em></td>
<td></td>
</tr>
<tr>
<td><em>If yes, will they be encrypted?: Yes</em></td>
<td></td>
</tr>
<tr>
<td>f.</td>
<td>After the research</td>
</tr>
<tr>
<td></td>
<td>Where will the data be stored? Digital data archived onto UCL network. Paper records containing personal data in lockable cabinets or offices with controlled access, when not under the direct supervision. Case notes will be kept in a written journal and stored securely in locked filing cabinet at UCL.</td>
</tr>
</tbody>
</table>
How long will the data and records by kept for and in what format? For five years after submission of PhD – tentative 2019 - The questionnaire data will be stored digitally. Field note encrypted on IOE laptop or written format.

Will data be archived for use by other researchers?  
* No ☒
* If yes, please provide details.

**Section 8  Ethical issues**

Are there particular features of the proposed work which may raise ethical concerns or add to the complexity of ethical decision making? If so, please outline how you will deal with these.

It is important that you demonstrate your awareness of potential risks or harm that may arise as a result of your research. You should then demonstrate that you have considered ways to minimise the likelihood and impact of each potential harm that you have identified. Please be as specific as possible in describing the ethical issues you will have to address. Please consider / address ALL issues that may apply.

*Ethical concerns may include, but not be limited to, the following areas:*

<table>
<thead>
<tr>
<th>Methods</th>
<th>International research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling</td>
<td>Risks to participants and/or researchers</td>
</tr>
<tr>
<td>Recruitment</td>
<td>Confidentiality/Anonymity</td>
</tr>
<tr>
<td>Gatekeepers</td>
<td>Disclosures/limits to confidentiality</td>
</tr>
<tr>
<td>Informed consent</td>
<td>Data storage and security both during and after the research (including transfer, sharing, encryption, protection)</td>
</tr>
<tr>
<td>Potentially vulnerable participants</td>
<td>Reporting</td>
</tr>
<tr>
<td>Safeguarding/child protection</td>
<td>Dissemination and use of findings</td>
</tr>
<tr>
<td>Sensitive topics</td>
<td></td>
</tr>
</tbody>
</table>
1. We are aware that the randomisation process undertaken by SHU will involve 50 schools that will be unable to undertake the intervention. We are addressing the issue by ensuring that control schools will do the CPD and consequent intervention in the following year.

2. Informed consent will be obtained from teachers as per the normal practice in the participating schools.

3. All data will be anonymised and individual questionnaires etc will be confidential and anonymised.

4. Data storage will be securely protected on the LKL development server, not in the cloud.

5. All reporting and dissemination will be anonymised in relation to individuals, teachers and schools.

Extract from Evaluation Protocol document

Participants and recruitment

Recruitment will begin during the design phase with the aim of identifying all schools by March 15th 2015. This will allow for signing of Memorandum of Understanding (MOUs), collection of pupils.

Schools will need to provide the following as a condition of being entered into the randomisation:

- MoU signed by Head. The MoU will include details of both IoE and SHU evaluation activities as well as information the school will be expected to supply
- Information on who would be attending the PD events if allocated to the intervention group
- Summary information on any previous use of Scratch programming, or engagement with Beaver tests (June 2015)

Section 9 Further information
Outline any other information you feel relevant to this submission, using a separate sheet or attachments if necessary.

This ethics approval is requesting an extension to the ethical approval which was submitted and approved in 2015 as part of my PhD study.

Tue 16/06/2015 14:55

Dear Piers

I am writing to confirm that ethics approval has been granted by the UCL Institute of Education for your doctoral research project entitled: “Collaborative Professional Development for ScratchMaths”

This ethics approval has been granted from 14 May 2015.

I wish you all the best for your forthcoming research.

Best wishes, Hazel

Hazel Croft
MPhil/PhD programme administrator
UCL Institute of Education
20 Bedford Way
London WC1H 0AL
### Section 10  Attachments

Please attach the following items to this form, or explain if not attached

<table>
<thead>
<tr>
<th>Item</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Information sheets and other materials to be used to inform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>potential participants about the research, including approach</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>letters</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>b. Consent form</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**If applicable:**

c. The proposal for the project                                      |     |    |
d. Approval letter from external Research Ethics Committee            |     |    |
e. Full risk assessment                                               |     |    |

### Section 11  Declaration

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have read, understood and will abide by the following set of</td>
<td></td>
</tr>
<tr>
<td>guidelines.</td>
<td>X</td>
</tr>
</tbody>
</table>

BPS [ ] BERA [ ] BSA [ ] Other (please state) [ ]

I have discussed the ethical issues relating to my research with my   |     |    |
supervisor.                                                         | X   |    |

I have attended the appropriate ethics training provided by my course. |     |    |

I **confirm that to the best of my knowledge:**

The above information is correct and that this is a full description of the ethics issues that may arise in the course of this project.

Name | Piers Saunders
Date | 18th May 2016

Please submit your completed ethics forms to your supervisor.

Notes and references
Professional code of ethics
You should read and understand relevant ethics guidelines, for example:
or
British Educational Research Association (2011) *Ethical Guidelines*
or
British Sociological Association (2002) *Statement of Ethical Practice*
Please see the respective websites for these or later versions; direct links to the latest versions are available on the Institute of Education [http://www.ioe.ac.uk/ethics/](http://www.ioe.ac.uk/ethics/).

Disclosure and Barring Service checks
If you are planning to carry out research in regulated Education environments such as Schools, or if your research will bring you into contact with children and young people (under the age of 18), you will need to have a Disclosure and Barring Service (DBS) CHECK, before you start. The DBS was previously known as the Criminal Records Bureau (CRB). If you do not already hold a current DBS check, and have not registered with the DBS update service, you will need to obtain one through at IOE. Further information can be found at [http://www.ioe.ac.uk/studentInformation/documents/DBS_Guidance_1415.pdf](http://www.ioe.ac.uk/studentInformation/documents/DBS_Guidance_1415.pdf)

Ensure that you apply for the DBS check in plenty of time as will take around 4 weeks, though can take longer depending on the circumstances.

Further references
The [www.ethicsguidebook.ac.uk](http://www.ethicsguidebook.ac.uk) website is very useful for assisting you to think through the ethical issues arising from your project.

This text has a helpful section on ethical considerations.

This text has useful suggestions if you are conducting research with children and young people.

A useful and short text covering areas including informed consent, approaches to research ethics including examples of ethical dilemmas.
If a project raises particularly challenging ethics issues, or a more detailed review would be appropriate, you may refer the application to the Research Ethics and Governance Administrator (via researchethics@ioe.ac.uk) so that it can be submitted to the Research Ethics Committee for consideration. A Research Ethics Committee Chair, ethics representatives in your department and the research ethics coordinator can advise you, either to support your review process, or help decide whether an application should be referred to the Research Ethics Committee.

*Also see* ‘when to pass a student ethics review up to the Research Ethics Committee’: [http://www.ioe.ac.uk/about/policiesProcedures/42253.html](http://www.ioe.ac.uk/about/policiesProcedures/42253.html)

<table>
<thead>
<tr>
<th><strong>Reviewer 1</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisor name</td>
</tr>
<tr>
<td>Supervisor comments</td>
</tr>
<tr>
<td>Supervisor signature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Reviewer 2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisory committee/course team member name</td>
</tr>
<tr>
<td>Advisory committee/course team member comments</td>
</tr>
<tr>
<td>Advisory committee/course team member signature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Decision</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date decision was made</td>
</tr>
<tr>
<td>Decision</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Recording

Recorded in the student information system

Once completed and approved, please send this form and associated documents to the relevant programme administrator to record on the student information system and to securely store.

Further guidance on ethical issues can be found on the IOE website at http://www.ioe.ac.uk/ethics/ and www.ethicsguidebook.ac.uk
Appendix G

G.1 Extract from the SM Memorandum of understanding

ScratchMaths project - Memorandum of Understanding

The ScratchMaths project is a randomised control trial that seeks to establish if the learning of computer programming in Scratch can improve not only computational thinking but also mathematics performance at Key Stage 2. It runs from late 2014 to the middle of 2017 and if successful, will subsequently be rolled out across the country. Please read through the following information and sign and return one copy of this document if you wish to join the project.

We hope that your school will take the opportunity to be one of 100 schools in England to participate in this innovative project based at the UCL Institute of Education and supported by the Education Endowment Foundation (EEF). Project partners include Sheffield Hallam University, London Connected Learning Centre (CLIC) and The National Association for the Advancement of Computer Education (Naace).

This project will produce materials and offer professional development that is aligned with the Computing Curriculum (Y5) and the Mathematics Curriculum (Y5s 5 and 6). It will aim to boost mathematics scores at KS2 by approaching some of the mathematics involved through creative programming in Scratch.

Participating schools will normally be two-form entry. Schools will also need to meet the technical requirements set out in the memorandum of understanding.

The project is a randomised control trial – similar to those used in medicine to test the effectiveness of a new treatment. The 100+ schools selected will be randomly split into 50 treatment schools and 50 control schools in March 2015.

Involvement in the CPD and delivery of the interventions will be staggered.

Treatment schools will

- receive two specially designed curriculum-aligned interventions for Y5s 5 and 6 in computational thinking and mathematical thinking, which include free student materials and teacher guidance using MIT’s Scratch software (http://scratc.hmit.edu). In addition we will provide two days of CPD, separated by a couple of weeks, in each of the summer terms 2015 and 2016 after KS2 testing.
- receive free CPD for teachers who will be teaching the interventions (computational thinking in Y5 and ScratchMaths in Y6).
- be invited to participate in an online teacher community for mutual support and advice.
- be required to share Y5 project resources with other schools (in year one of the project) and Y6 project resources (in year two)

Control schools will receive free access to Y5 ScratchMaths materials and training in computational thinking in summer 2016. They will receive all of the materials for Y6, in 2017. They will be invited to participate in an online community in summer 2016.

All schools will receive feedback on the outcomes of the study to inform future practice.

Sheffield Hallam University will be conducting the analysis of the effects of the intervention. To take part, schools will need to provide Sheffield Hallam with data on teachers who will be taking part as well as the
Unique Pupil Numbers (UPNs) for all Year 5 pupils in the school. Schools will also need to inform parents and give them the choice for their children to opt-out.

To see if the programme is effective, Sheffield Hallam University will retrieve KS1 scores for the Y5 pupils from the National Pupil Database. In Summer 2016, the Y5 pupils will take an on-line test of computational thinking arranged by Sheffield Hallam. Support will be given to schools to administer this. Test taking can be staggered and schools will be able to choose when the pupils will take the test, within a given time period. At the end of the trial, Sheffield Hallam will retrieve KS2 data from the National Pupil Database.

<table>
<thead>
<tr>
<th>Application to apply</th>
<th>Treatment schools</th>
<th>Control schools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer '15</td>
<td>CPD computational thinking (Y5)</td>
<td>Complete MOU. Provide pupils UPN to SHU</td>
</tr>
<tr>
<td>Autumn '15 /Spring '16</td>
<td>Computational thinking intervention (Y5) online teacher survey</td>
<td>(ongoing school activities for Y5 computing) online teacher survey</td>
</tr>
<tr>
<td>Summer '16</td>
<td>CPD ScratchMaths Computational thinking test (Y5) online teacher survey</td>
<td>CPD computational thinking Computational thinking test (Y5) online teacher survey</td>
</tr>
<tr>
<td>Autumn '16 /Spring '17</td>
<td>CPD ScratchMaths intervention (Y6) online teacher survey</td>
<td>Computational thinking intervention (Y5) online teacher survey Computational thinking test (Y5)</td>
</tr>
<tr>
<td>Summer '17</td>
<td>Key Stage 2 Mathematics test as normal</td>
<td>Key Stage 2 Mathematics test as normal</td>
</tr>
</tbody>
</table>

100 schools will be selected for the trials and randomly split into 50 treatment schools and 50 control schools. Prior to the trial starting ALL schools will...

Technical requirements

- ensure adequate Internet connectivity is available for Y5 and Y6 pupils at the time of the interventions.
- ensure enough machines are available (one between two at least) for the Y5 and Y6 pupils at the time of the interventions.
- to run Scratch 2.0 online with the whole group/class of pupils in parallel, you will need a relatively recent web browser (Safari, Chrome 7 or later, Firefox 4 or later, or Internet Explorer 8 or later) with Adobe Flash Player version 10.2 or later installed. Scratch 2 is designed to support screen sizes 1024 x 768 or larger.
- Scratch 2.0 does not work on iPads and similar devices, so we advise testing Scratch on the computers to be used before agreeing to take part in the project.

NOTE: The project team has received written consent from MIT’s Scratch team that each school will be allowed to set up individual Scratch accounts for each of their participating pupils.

Data

- provide information on request about the school and two teachers who will be involved in the project. This includes teachers’ attendance at CPD, current activities related to programming and the National Curriculum for Computing; information on any other use of Scratch programming in the school; information on any testing procedures for computing. A link to a short online form will be sent to each school for completion.
## Appendix H

### H.1 Visit schedule

<table>
<thead>
<tr>
<th>School</th>
<th>Teacher</th>
<th>(Module) Microworld</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(Module 4) Place value</td>
</tr>
<tr>
<td>Local Area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bradford</td>
<td>A Laura</td>
<td>03/11/2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>03/11/2016</td>
</tr>
<tr>
<td>Merseyside</td>
<td>B Matt</td>
<td>25/11/2016</td>
</tr>
<tr>
<td></td>
<td>C Katherine</td>
<td>18/11/2016</td>
</tr>
<tr>
<td>North London</td>
<td>D Mary</td>
<td>02/12/2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>02/12/2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12/10/2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>02/02/2018</td>
</tr>
<tr>
<td></td>
<td>E Rina</td>
<td>09/11/2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sally</td>
<td>22/03/2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>05/06/2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21/06/2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28/06/2017</td>
</tr>
<tr>
<td></td>
<td>F Priya</td>
<td>16/03/2017</td>
</tr>
<tr>
<td>South London</td>
<td>G Elizabeth</td>
<td>22/05/2017</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23/05/2017</td>
</tr>
<tr>
<td></td>
<td>H Betty</td>
<td>28/02/2017</td>
</tr>
<tr>
<td>Blackburn</td>
<td>I Nancy</td>
<td>29/03/2017</td>
</tr>
<tr>
<td></td>
<td>Christine</td>
<td>29/03/2017</td>
</tr>
<tr>
<td>Somerset</td>
<td>J Matt</td>
<td>10/11/2016</td>
</tr>
</tbody>
</table>

Table 10.2 ScratchMaths teachers observed across each Microworld. Each observed lesson length was between 60 and 100 minutes.
Appendix I

I.1 Scratch overview and glossary

Definitions and glossary text adapted from the Scratch wiki (MIT, 2022).

Scratch is a programming language that makes it easy to create interactive art, stories, simulations, and games – and to share those creations online. It is freely available in more than 70 languages at https://scratch.mit.edu/

![Scratch stage with a sprite on it](image)

Figure 10.2 An image of the Scratch stage with a sprite on it

A Beetle sprite is shown in Figure 10.2 above. A sprite is an object or character in Scratch that can be programmed to perform actions based on scripts in a project using blocks. Each sprite has its own scripts and costumes and can move on its own.
A project is a creation made in Scratch. In the context of the thesis, a project is a microworld containing custom created sprites, costumes and readymade blocks. [See microworld definition on page 9 and Chapter 4 – The ScratchMaths curriculum and microworlds].

A script is a collection or stack of blocks that all interlock with one another.

A costume is one out of possibly many "frames" or alternate appearances of a sprite. Sprites can change their look to any of its costumes.

The stage is the background of the Scratch project, it can have scripts, backdrops (costumes), similar to a sprite. However, sprites cannot move behind the stage. The stage size is always 480x360 pixels with the origin in the centre and represents a Cartesian coordinate system. Sprites can have a coordinate position such that the X position can range from 240 to -240, where 240 is the rightmost a sprite can be and -240 is the leftmost, and the Y position can range from 180 to -180, where 180 is the highest it can be and -180 is the lowest it can be.

A broadcast is a message that is sent through the Scratch program using the Broadcast () block and activating scripts with the matching When I Receive () block. Broadcasts are used to trigger specific scripts.

The move () block is a motion block, it moves the sprite forward the specified number of steps in the direction that its facing. Each step is equal to one-pixel length.

The turn right () degrees and turn left () degrees are motion blocks. The block respectively turns the sprite’s direction in the specified amount of degrees clockwise or anticlockwise.