

Physical Computing with Plug-And-Play Toolkits: Key Recommendations for Collaborative Learning Implementations

Abstract

Physical computing toolkits have long been used in educational contexts to learn about computational concepts by engaging in the making of interactive projects. This paper presents a comprehensive toolkit that can help educators teach programming with an emphasis on collaboration, and provides suggestions for its effective pedagogical implementation. The toolkit comprises the Talkoo kit with physical computing plug-and-play modules and a visual programming environment. The key suggestions are inspired by the results of the evaluation studies which show that children (aged 14 to 18 in a sample group of 34 students) are well motivated when working with the toolkit but lack confidence in the kit's support for collaborative learning. If the intention is to move beyond tools and code in computer education to community and context, thus encouraging computational participation, collaboration should be considered as a key aspect of physical computing activities. Our approach expands the field of programming with physical computing for teenage children with a focus on empowering teachers and students with not only a kit but also its appropriate classroom implementation for collaborative learning.

Keywords

Collaborative learning; education; motivation; physical computing; programming; toolkit

1. Introduction

While physical computing toolkits have long been used in educational contexts for learning about computational concepts by engaging in the making of interactive projects [1][26], their implementation in the classroom remains challenging. Common hardware components require skilled tutors and continuous guidance, especially for novice users. Struggling with hardware assembly and textual programming syntax prevents students from becoming deeply involved in computational concepts [2]. Appropriate kit designs that allow children to build projects within a short time frame without having to struggle with bad connections or programming syntax could help overcome such issues. However, the uptake of these technologies in education remains a complex process. Much research on the use of physical computing toolkits in the context of practice-based learning has highlighted the important potential to improve students' subject-specific knowledge and understanding, their higher tier skills such as their practical, and metacognitive skills, as well as their motivation and attitudes [3][4][5]. However, a clear gap exists between this potential and what actually happens when these kits are used in educational settings. Although many computing educators attempt to integrate some form of physical computing into their teaching, teachers receive little systematic support for applying such teaching approaches. Students are often found seated or working in some form of group using materials, such as physical computing materials, but many of these groupings with hands-on activities may not actually lead students to achieve the expected learning outcomes [6][7]. To achieve learning outcomes, not only those associated with knowledge acquisition but also students' skill development, there is a need for educational resources that will help teachers to apply learning activities with appropriate pedagogies [8].

The overall goal of our research is to design an integrated physical computing kit that motivates students across educational contexts (high school and university levels) to create and program collaboratively through interactive projects in class. In this paper, we present the kit we designed as well as the results of its evaluation. The evaluation results lead us to further ask what kind of additional support might students require in a high school context for effective collaborative learning. As a consequence, we present an integrated approach with a novel toolkit to ease physical computing for young users. The kit we developed was evaluated in two steps. The first evaluation (“field test”) focused on the suitability of the hardware parts and software for high school students. In the second evaluation step (“user trials”), students’ motivation and (perceived) collaboration were evaluated as part of in-depth user trials with students at high school and university levels. After the evaluations, the kit was complemented with key suggestions for implementation, based on our literature research. These key recommendations are designed to help teachers orchestrate physical computing activities in a way that is pedagogically appropriate to improving students’ social and cognitive skills for learning collaboratively.

In this paper, we purposefully avoid focusing specifically on computational thinking with a limited definition; instead, we utilise the concept of computational participation [9]. As defined by Wing [10], computational thinking is taking an approach to solve problems, design systems and understand human behaviour that draws on concepts fundamental to computing. Computational thinking thus shares several key characteristics with other types of analytical thinking including mathematical, engineering, and scientific thinking. As argued by Wing [10], computational thinking “shares with mathematical thinking in the general ways in which we might approach solving a problem. It shares with engineering thinking in the general ways in which we might approach designing and evaluating a large, complex system that operates within the constraints of the real world. It shares with scientific thinking in the general ways in which we might approach understanding computability, intelligence, the mind and human behaviour” (p. 2). Based on these arguments, Kafai extended the idea of computational thinking to computational participation stressing the importance of collaborative skills [9]. Hence, we focus on the value of collaborative learning while engaging with the fundamental concepts of physical computing that are introduced by the Talkoo kit.

Organized chronologically, this paper presents the development, evaluation and key suggestions for implementation of the toolkit, which reflects our research and development process. Section 2 presents the background and related work on challenges of implementing physical computing kits in secondary education. Section 3 presents the Talkoo kit and Section 4 reports on the two-step evaluation in different educational contexts. In Section 5, we offer pedagogically grounded key suggestions on how to implement Talkoo in high school classrooms, and we conclude by discussing the results and further research of toolkit and its potential curriculum integration.

2 Background and related work

Physical computing adds state-of-the-art topics to the computer science curriculum, such as sensing and control or ubiquitous and embedded systems, and is suited to introducing programming concepts in a meaningful way [11]. Activities with such toolkits are linked to developing students’ practical skills, fostering motivation and enhancing social and cognitive skills of learning by doing. However, implementing physical computing in the classroom imposes challenges on students and teachers; the technology design and contextual conditions typically impair the successful uptake and adoption and thus fail to achieve the desired learning objectives.

2.1 Physical Computing Technology Designs in Education

Physical computing toolkits for beginners employ different types of technology designs that affect their appropriateness for education. While popular boards such as the native Arduino¹ are low-cost and have the support of a huge community, they require breadboards, wires and soldering, and the mastering of a professional programming language which can be intimidating to learners, making realizing and iterating designs a challenge [12]. The challenges of electronics and programming might inhibit learners' motivation and engagement with the activity. Further, the activities required to get a project working might restrict the focus on computational concepts, especially in school contexts where time is limited by a schedule and workload is always at the high end. As a consequence, educational designs that focus more on usability and cognitive aspects are required [2].

To be able to work with toolkits in class, teachers need to have a solid background in electronics and programming. Although the average class size or students-per-teacher ratio in Europe is about 21 students [13], project-based physical computing activities require continuous support. Therefore, to establish physical computing in regular school settings and schedules, kit designs need to be made manageable for teachers with little previous knowledge and allow students to build projects together within a short time frame, without having to overcome bad connections and circuits. To enable the next class to work with the kits with minimum delay, hard and software setups should be reusable.

Attempts have been made to focus on computational concepts and activities beyond electronics and to avoid an overload of electrical engineering, usually by providing higher-level electronic components that hide their inner electronics. Toolkits such as (basic) LittleBits [14], Tinkerbots,² Cubelets³ and SnapCircuits⁴ allow for building electronic artefacts or circuits by plugging together block-like components to build program-like structures that do not involve any programming. While this method introduces basic computational concepts, it is not possible to advance much further. Bloctopus implements physical blocks and visual programming with components to be programmed via a Web interface, demonstrating the feasibility of a modular system for prototyping (but not primarily educational) purposes [15], and while LEGO's kits are pluggable, they are rather specialised on robotics⁵ or the primary school level.⁶ Single-board designs such as the BBC microbit⁷ make it easy to get started without assembling parts and can be easily reused for other purposes. However, integrated sensors (and associated concepts) may remain black boxes, and there is little flexibility in terms of extending the variety and number of connected components.

On the other hand, visual programming languages (VPLs) can make programming more accessible for children and novices [16] by translating graphical blocks into textual code. Booth and Stumpf [17] argued that (adult) learners using VPL have an easier time adapting code from other projects and have a more positive experience than learners using traditional text-based interfaces. Programming microcontrollers (especially in the beginning) involves dealing with incoming (input) sensor signals, doing some operations on their values and

¹ Arduino: <https://www.arduino.cc/> (retrieved 01/02/2017).

² Tinkerbots: <https://www.tinkerbots.net> (retrieved 01/02/2017).

³ Cubelets Six Robot Construction Kit: <http://www.modrobotics.com/cubelets/cubelets-six/#kit-includes> (retrieved 01/02/2017).

⁴ Snap Circuits: <http://www.snapcircuits.net> (retrieved 01/02/2017).

⁵ LEGO Mindstorms: <https://www.lego.com/en-us/mindstorms> (retrieved 01/02/2017).

⁶ LEGO WeDo: <https://education.lego.com/en-us/products/lego-education-wedo-construction-set/9580> (retrieved 01/02/2017).

⁷ BBC micro:bit: <https://www.microbit.co.uk> (retrieved 01/02/2017).

outputting them to actuators. Especially for beginners, this interplay between inputs and outputs makes tinkering with microcontrollers rewarding and appealing. While existing visual programming environments for physical computing follow a procedural, text-like paradigm⁸, it makes sense to think about new ways of programming to support collaborative activities by reflecting the structures of hardware and computing operations in program code and by giving clear feedback about the state of hardware components in real time.

2.2 Uptake of Technology in Classrooms

Despite appropriate tool design, the uptake of technology in education is a complex process with multiple stages not only affected by technology constraints. NESTA recently broadly addressed this issue in a report, which concluded the following ways that the context in which digital technology is deployed might need to change to achieve better educational outcomes:

- Focus on activity, not technology,
- Support the teacher, and
- Link homes schools and the community [18].

A more recent comprehensive literature review of research on technology innovations in STEM (Science, Technology, Engineering, and Mathematics) education, its uptake, the factors affecting the uptake and the challenge of understanding their effects categorises the technology landscape in nine categories, which included experiential learning through mixed reality technology and collaborative problem solving through communication [19]. The review concluded that “there is evidence to show that use of this equipment is not widespread and that it is often used in limited ways” [20][21][19]. Factors that have been identified by multiple studies and reports to be lacking include teacher confidence, access, training time and technical reliability. In addition, although some evaluation studies report on teachers’ uptake of technology-enhanced teaching methods and programmes, few studies have directly addressed the uptake within the large-scale development and implementation programmes of digital learning technologies. More recently, projects such as Cornerstone Maths in England [22] and SimCalc in the USA [23] have begun to address this research gap, highlighting the need for a blended design that incorporates technology enhanced curriculum units, teacher professional development and on-going teacher/community support networks within and across individual schools.

2.3 Implementation is Key for Technology Adoption in Education

Within the general education literature, the process of scaling any educational innovation have been theorised differently. In a seminal paper, Coburn [24] develops a concept of scale in relation to four interrelated dimensions: depth, sustainability, spread and a shift in reform ownership. She challenges the view of scaling-up in terms of only quantitative measures (numbers of teachers, schools, educational districts etc.) to include the “depth of implementation and a shift in reform ownership” ([24] p. 4). However, Coburn makes no particular reference to educational technology innovations.

The 2010 OECD Report “Inspired by Technology, Driven by Pedagogy” addresses this gap, and Hung, Lim and Huang [25] offer a framework for considering the scaling of such innovations within STEM subjects, including computer science, informed by their systematic analysis of a programme of research at NIE Singapore. Hung and colleagues challenge the product-oriented view by stating, “The process (rather than factors) in which pre-adopters make transitions through their participatory involvements in the community has not generally

⁸ E.g. Snap for Arduino: <http://http://snap4arduino.org> (retrieved 01/02/2017); micro:bit editors: <http://microbit.org> (retrieved 01/02/2017).

been the focus of investigation. An often implicit assumption of this view is that these innovation ‘products’ are replicable en masse without undergoing (and, without the need to undergo) significant change from the original” ([25] p. 90).

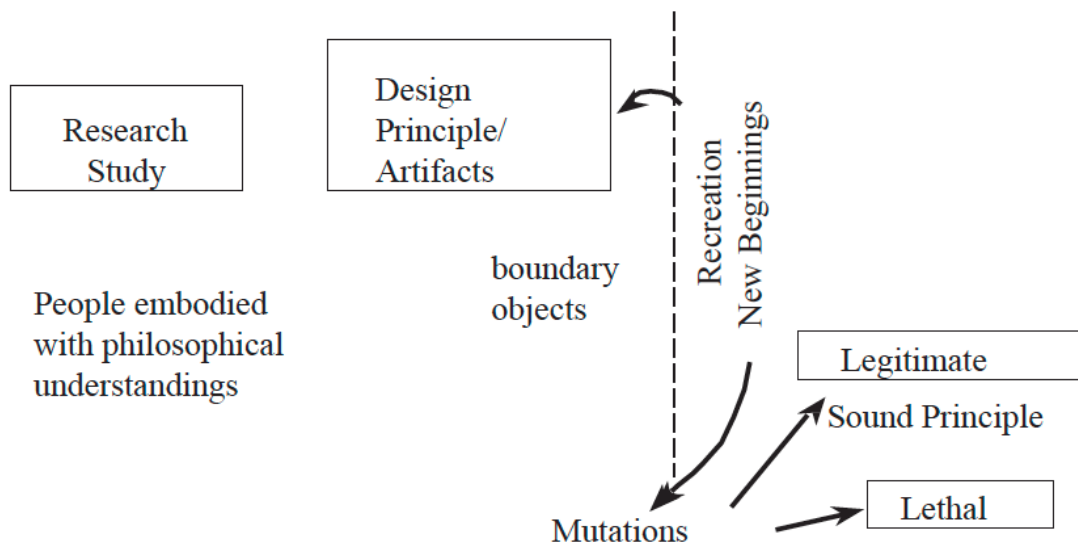


Fig. 1. Framework for translation and extensions/scaling innovations ([25] p.93).

Hung and colleagues offer a framework (Fig. 1) which considers scaling in terms of legitimate and lethal mutations of the original design principles/artefacts, thus foregrounding the fact that mutations are an inevitable, desirable and healthy component of scaling.

Focusing our attention on the particular physical computing context for the purposes of this paper, the research evidence is only recently emerging as different jurisdictions implement or re-emphasize computer science curricula and the increased availability of cheap or free physical computing devices (as introduced in Section 2.1).

Central to the scaling of computer science education is the need to design and implement professional learning opportunities to up-skill the teaching workforce to meet the demands of new curricular, examinations and societal expectations [26]. Effective collaborative learning is one such demand that is key to effective implementation. In practice, activities with physical computing kits are generally implemented as group work. As a consequence, in the process of developing appropriate technologies and adopting them into educational contexts, we see the support of collaboration as essential to its effective implementation. Further, tools and activities need to be motivating for students and teachers to keep them engaged. In the next section, we present our attempts to design the Talkoo toolkit to meet these requirements.

As explained here (and will be confirmed by our evaluation), the adoption and scaling of technology goes beyond tool design. We provide key suggestions for successful implementation of the kit in Section 5 to provide a potential best-practice implementation guide of the Talkoo kit (and potentially other similar physical computing toolkits) with a focus on collaborative learning and engagement.

3 The Talkoo Toolkit

We developed the Talkoo kit to introduce physical computing into school with an emphasis on increasing student motivation and collaboration and decreasing teachers’ workload. Talkoo comprises physical computing plug-and-play modules, a visual programming environment and prototyping material. The kit was designed to allow beginners in secondary

and tertiary education to get started with building electronics while avoiding any possible errors produced by miswiring. At the same time, advanced users can take advantage of this Arduino-based toolkit when trying to quickly prototype interactive systems.

The Talkoo kit, which implements a plug-and-play concept with real-time feedback, is made of a series of smart modules that can be simply plugged to one another in any order without soldering or breadboards. When modules are connected to a computer, they automatically appear on the screen of a visual programming environment ready to be programmed by drawing virtual connections. The physical modules are “mapped” on the screen and interact in real time with the programming environment to give feedback on hardware performance.

Technically speaking, each hardware module is a microcontroller board with a single sensor or actuator connected to it that makes use of a bus communication protocol to acquire information and send it to and receive it from a computer. From a user viewpoint, it is a system of small PCBs (2.5 x 2.5 cm), each with a sensor or actuator attached, and one hub module with a USB connection that operates at run-time. To build a project, the hub first needs to be connected to the computer followed by a chain of desired modules (Fig. 2). The modules are connected using small cables with directed connections. The order of the modules is independent of the programming. The name and common icon written on the backside of each module help distinguish it from another. Compared to toolkits with breadboard circuiting and tiny components, the module design facilitates building an artefact with multiple users (and hands) at the same time.

The hardware modules are represented virtually as blocks—interactive graphical elements—on the visual programming environment, i.e. one block stands for one sensor or actuator. This programming environment is simple and direct because it allows a direct connection between blocks (e.g. connecting a button to an LED) and the addition of some basic operations to be performed on the data, such as comparisons and timings, by placing logic blocks in between (Fig. 3). By mapping physical modules directly onto virtual blocks, the programming environment implements an intuitive event-driven paradigm. Actuator and sensor blocks have the same symbol as that printed on the corresponding module’s backside. The program does not need to be uploaded but runs immediately on the modules while the virtual blocks display the flow of incoming/outgoing values in real time and thus facilitates debugging. Users can focus on the programming logics instead of struggling with hardware issues. Compared to dense textual program code, the visual code is overseenable by groups of users gathering around.

The Talkoo plug-and-play design concept with real-time feedback and a clear visual programming language has several advantages. First, it can allow more time and space for the programming part because it directs students’ focus towards applying computational concepts beyond syntax or circuit issues. Second—and most importantly for our research focus—it can reduce frustration due to fewer problems with miswiring, bad connections or getting lost in programming code. This concept also allows more time and space for collaboration. Further, the design of the modular self-contained components and the overseenable programming interface support joint building, programming and debugging of projects.

Another characteristic of the Talkoo programming environment is its connection to the PELARS Learning Analytics System (LAS) for gathering information pertaining to which modules are connected and when [27]. This feature has been used to capture information on the usage of hardware modules and software blocks in the evaluation field test (Section 4).

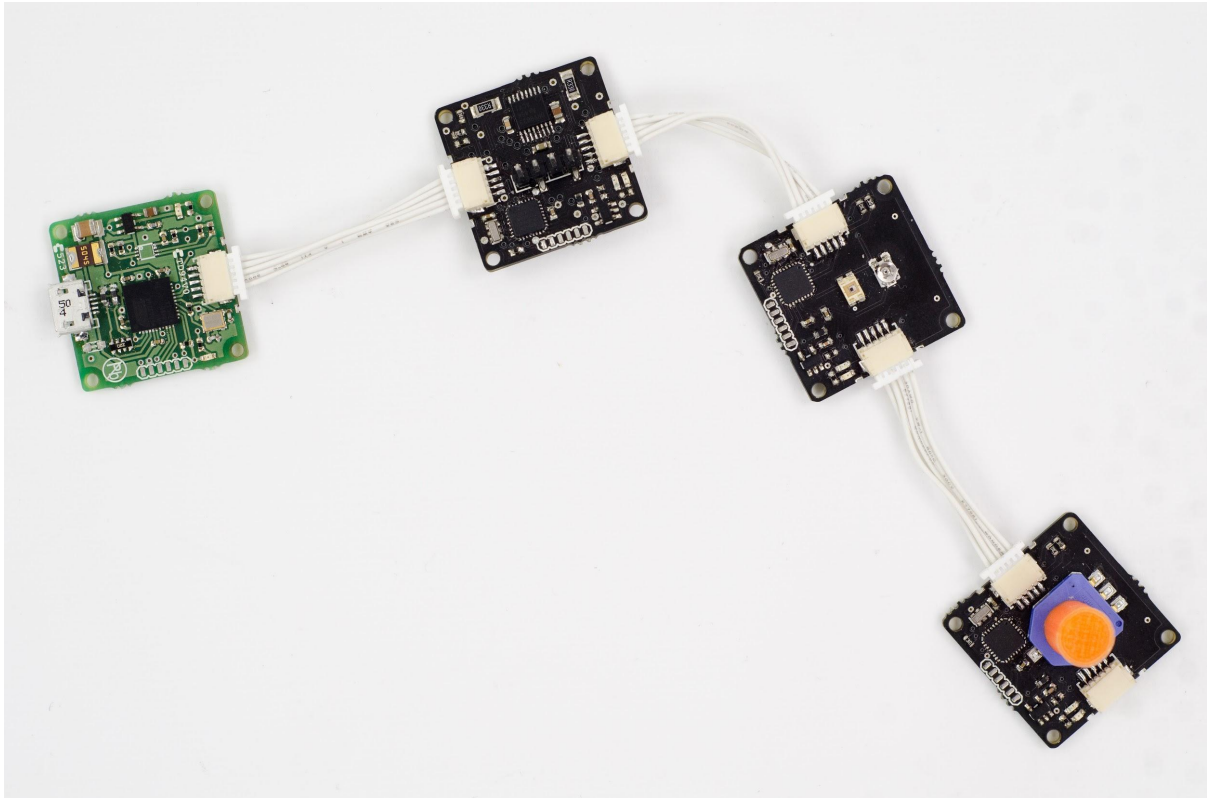


Fig. 2. Talkoo modules (hub, motor, colour sensor, potentiometer) (with credit to Arduino Verkstad AB, photographer: Mike Ericsson, 2015 CC-SA-BY-NC).

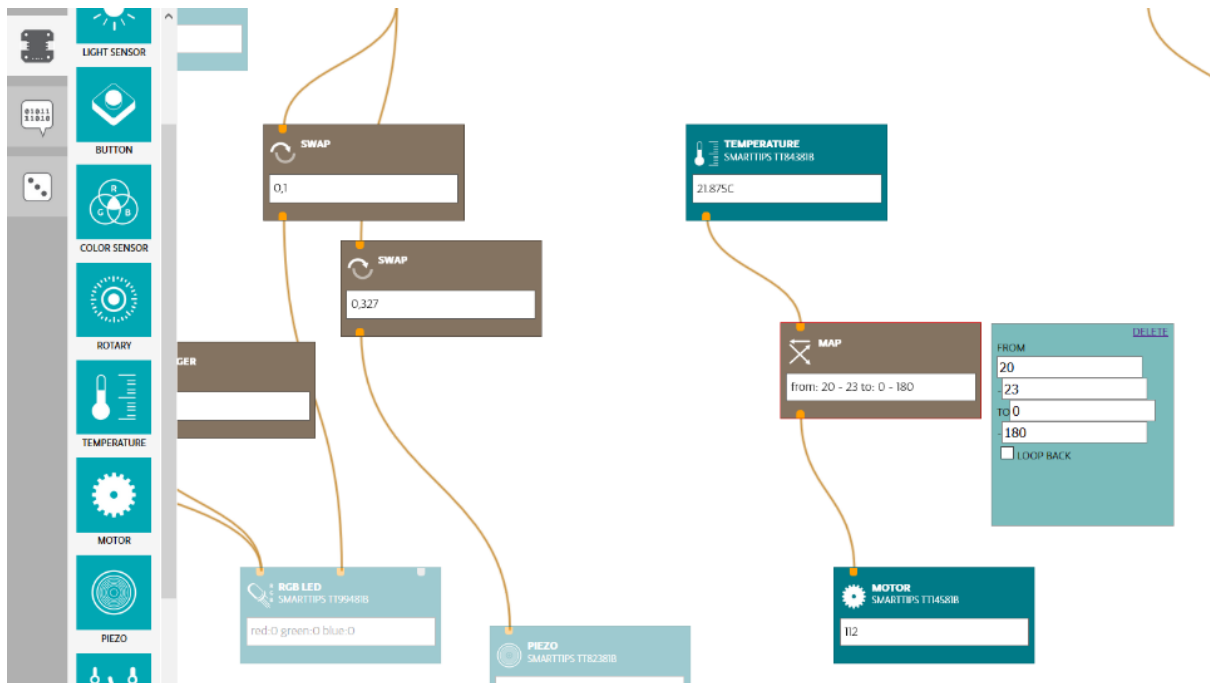


Fig. 3. Sample program with Talkoo's visual programming environment (with credit to Arduino Verkstad AB, photographer: Mike Ericsson, 2015 CC-SA-BY-NC).

4 Evaluation

Talkoo was evaluated during two evaluation phases: a field test and user trials. The goal of the field test was to determine whether high school students could create and program

interactive projects using Talkoo. In other words, the field test evaluated whether the kit minimised usability issues for further evaluation. The goal of the user trials was to find out how the users perceived the kit's support for collaboration and motivation.

The evaluation work discussed in this article was conducted within the European project "Practice-based Experiential Learning Analytics Research and Support" (PELARS)⁹. The overall aim of the project was to develop learning analytics tools for hands-on, open-ended STEM learning activities using physical computing. The learning contexts investigated are STEM education in high school, engineering and interaction design at universities. The system included specially created furniture (round table with monitor and whiteboard) with an integrated LAS including tracking hands, faces and other objects (Fig. 4). The Talkoo kit was an integral part of the project. Besides the toolkit, the learners and observers used mobile devices to capture multimedia data (text, images and video) to self-document the learning activities. Overall, the PELARS project has developed an intelligent system for collecting activity data for diverse learning analytics (with data-mining, reasoning and visualisations) and active user-generated material and digital content (mobile tools) from practice-based activities [28]. The Talkoo toolkit was used and evaluated in combination with the PELARS LAS. Besides feedback about hardware kit modules and programming, the LAS collected data from multimodal resources for exploratory research purposes; however, these data cannot yet be used to judge the level of collaboration reliably because they are still under investigation [29]. The motivation and collaboration data presented in this paper were measured using Likert-type questionnaires based on students' self-declared data.

PELARS was funded by the European Commission whose ethical guidelines were followed at the European and national levels for consent from adults and children. Informed consent was obtained from all participants and their guardians in conjunction with the respective institutions. In the following, we report on the two evaluation settings that gave insights into the suitability of Talkoo for high school contexts.

⁹ PELARS project website: <http://pelars.eu> (retrieved 01/02/2017).



Fig. 4. High school students using the PELARS system with Talkoo hardware.

4.1 Evaluation Phases

The PELARS project adopted a design driven approach that brings together design-based educational research efforts [30] and human-centred interaction design [31]. An early version of the Talkoo kit with the PELARS system was tested at an event (Ars Electronica U19) where designers and developers worked directly with learners to obtain feedback [32][33]. Over the next several months, the entire PELARS system, including Talkoo, went through minor revisions and development based on feedback from this event. These changes included new revisions of the programming environment to improve its technical performance, including a revision of the firmware of the hardware modules [33].

The new version was implemented for the user trials of the main project where learning analytics data were collected along with survey data, observations and interviews of the learners and teachers. Note that this article focuses on both high school students and university students. The PELARS project included both groups of students, and we chose to compare the experiences of the different level of students to understand what works best across the learning contexts and to compare the needs of teenage children with those of higher-level students. Table 1 shows an overview of the different approaches of the two evaluation phases described in the following subsections (those referred to in this paper are highlighted in *italics*).

Phases	Focus	Context	Students	Approach	Instruments	Users
Field Testing	Usability issues of Talkoo	Ars Electronica Festival	High school	Exploratory: observations, workshops, <i>data from PELARS on kit usage</i>	Workshop, participatory-design sessions, <i>data analytics</i>	14 in workshops 86 ad-hoc
User Trials	Collaboration, satisfaction with outcomes and motivation	Classrooms at formal institutions	High school and university	Mixed methods	Pre, in-situ, <i>post surveys</i> , data analytics, structured observations	57 university 20 high school in 44 group sessions

Table 1. Design phases, contexts and general approach. Approach and instruments reported in this article are set in italics.

4.1.1 Field Testing

Talkoo was tested at the “U19 Create Your World Exhibition”¹⁰ for young people at the Ars Electronica Festival in 2015. Participants were recruited in conjunction with specific class visits to the festival. The organizers of U19 supported the recruitment based on local schools. Fourteen children aged 14 to 18 participated in three workshops. Each of the workshops was shaped as “a design challenge,” where the participants received the task to protect a candy shop from fire, earthquake, robbery or other possible calamity. The three teams of the 14 high-school students were introduced to Talkoo in a 15-minute session, and the teams had 40 minutes to create prototypes of their solutions. The students could get help from designers and developers who were around. At the end, each group presented their ideas and explained how they implemented them, and a small jury (composed of four researchers from the project) decided on the winner. We also collected data from 76 more ad-hoc sessions involving high school students who came to the booth to explore and tinker with the toolkit. The overall goal of this field test was to gain insights into the understanding and usage of the hardware modules and the visual programming environment. At Ars Electronica, we generally had one helper per team of students.

We used data about the toolkit collected by the LAS to explore a measure of success rate at understanding and using the basic kit components and programming environment. Since part of the goal of the intervention was to understand how we could use the data from LAS, we collected data about how the students used the Talkoo toolkit. The PELARS system collected data from 100 sessions during the event including the three workshops. With a specific focus on the Talkoo kit, we broke the process of using the hardware and software down into three actions that needed to happen to ensure success: (1) connect the hardware module(s) to the hub (the block that connects the kit to the PC), (2) add the corresponding software visual block(s) to the sketch area on the display, and (3) link the visual block(s) to others on the screen and adjust the software parameters. To ensure success, the usage pattern needed to contain a complete action chain “hardware module—logical block—linking.” In the case of logical software blocks, only actions (2) and (3) were required. From the PELARS system, we tested seven hardware modules (sensors and actuators) and six software blocks for the visual programming.

This rough granular analysis showed a general success rate for the hardware modules; for instance, 68 users were successful in using the button hardware module and 13 failed (Fig. 5), and for the LDR sensor, only 26 out of 38 participants were successful. In terms of the

¹⁰ Ars Electronica “U19 Create Your World Exhibition”: <https://www.aec.at/u19/en/> (retrieved 20/09/2017).

software, the if-then visual programming block had 32 successes and seven failures. While this result does not necessarily mean that the users accomplished the goal of using a given block in the project, it is a good indication that users have understood the basic functioning logic of the kit and its blocks, thus providing the design team with insights into what to improve to maintain motivation. The primary required improvement was to improve the software performance. Our intention for analysing the overall data from the 100 sessions was to gain insights into the overall perspective of how users interacted with the software and hardware. However, because the PELARS system was in its early stages, we were unable to separate the data obtained from the workshops and the rest of the participants.

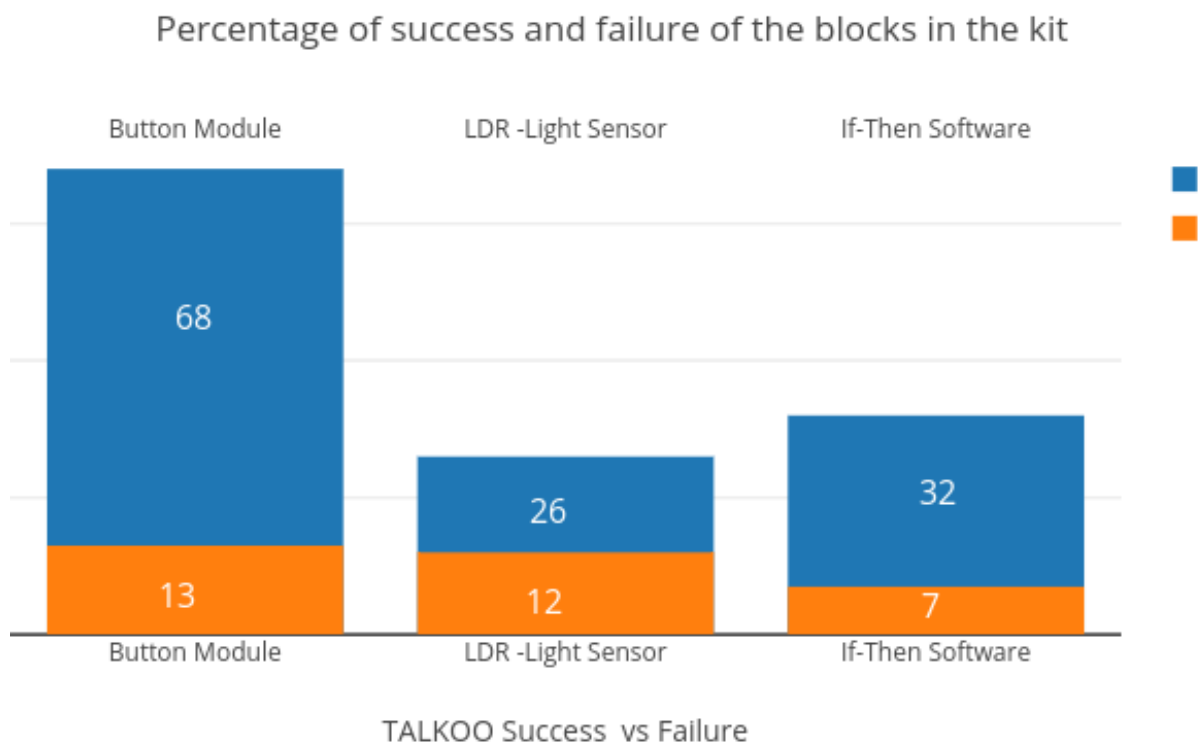


Fig. 5. Successes and failures (%) for selected blocks of the toolkit/programming environment.

4.1.2 User Trials

To evaluate phase 2, we conducted 44 user trials with 128 students (24 high school and 104 university level design and engineering) in groups of three. The high school trials took place in Spain and the UK, and the university trials were held in Romania, Denmark, Sweden and Ireland. The average age of the high school students was 15 years old, the design students' average age was 22, and the engineering students' average age was 20.

The procedure for the trials was a pre- and post-intervention survey to determine the students' motivation to use the elements of physical computing and to find out their experience with collaborative learning. During a 20- to 30-minute workshop, the students were introduced to the software and hardware elements of the toolkit and then given the challenge of designing an interactive toy within 60 to 100 minutes. The students could use Talkoo, crafting material and tools (scissors, cutter knives, cardboards, coloured papers etc.). The facilitators purposefully avoided giving any specific didactic guidance for the ideation processes and group work. One main facilitator ran the user trials, and when we ran

two or more sessions on multiple tables, we added an additional facilitator for support. After the session, the students were given a post-intervention questionnaire survey on the mobile devices to measure their perceived level of motivation and collaboration. The questionnaire items were devised based on self-determination theory [34] for motivation and the teamwork questionnaire (TWQ) [35] for collaboration. In this paper, we present the results from the second and third iterations of user testing that had complete data sets from the LAS (n=77).

From the post-intervention survey conducted during the user trials, we explored how motivated the students were to use the Talkoo kit (as part of the PELARS LAS) and how well they collaborated when working with the physical computing kit. The data were collected on a group level. The questionnaire items regarding motivation were measured using a five-point Likert scale, with 5 indicating a high level of motivation. The items asked how the students felt about their project and if they were pleased with their artefact and the experience of building. Table 2 shows that, overall, the students felt positive about the experience; 37 university students had a mean of 3.92 and an SD of 1.01, while the high school student groups had a mean of 4.25 and an SD of 0.89, suggesting that regardless of age, the students felt strongly positive about what they had built and accomplished with the Talkoo kit.

University		High School	
N=	37	N=	7
Mean	3.92	Mean	4.25
SD	1.01	SD	0.89

Table 2. Students' satisfaction with their projects (group based).

Table 3 shows the breakdown between the different learning contexts regarding how students felt about their group's collaboration using the Talkoo and the PELARS system. The overall mean was 3.36 with an SD of 1.29, which indicates that all students were positive towards that the PELARS system and they considered that the Talkoo kits made it easy to collaborate. However, based on the three different educational backgrounds and the two age groups, the responses differ slightly. The means obtained for the engineering students (4.32; SD=0.58) and design students (3.79; SD=0.81) indicated much more positive attitudes towards the ease of use that PELARS provided (Table 2) than those of the high school students (1.65; SD=0.81), thus indicating that high schools students felt less at ease with collaboration when using Talkoo. All data collected from the different groups of students were collected from the PELARS mobile system.

All Students		Engineering		Design		High School	
N=	77	N=	19	N=	38	N=	20
Mean	3.36	Mean	4.32	Mean	3.79	Mean	1.65
SD	1.29	SD	0.58	SD	0.81	SD	0.81

Table 3. Perception of collaboration when using the PELARS system (table and toolkits; individual based).

5 Adoption of the Talkoo Toolkit in Classrooms and Its Successful Implementation in Practice

5.1 Discussion of evaluation results

The data analysis of the field tests showed that the high school students gained a good understanding of the toolkit in an informal setting. The results from the event showed that

Talkoo is appropriate for teenagers and enables them to build interactive projects quickly and effectively. The results from the workshops, ad-hoc visiting and simple data collection provided details for refining the early version of the Talkoo kit.

However, the results from user trials are more nuanced. A detailed analysis of the user trials indicates that the university students perceived the kit to support collaboration and motivation (along with the entire system) more than did the high school students. However, an examination of how the students felt about their artefact after the activity shows that the high school students were more satisfied than university students with their artefact, which we consider an indication of their motivation. While the user trials investigated the entire PELARS experience, the students worked with the Talkoo kit, while the furniture, the mobile system, and the LAS remained in the background. Additionally, as the questionnaires were answered in groups, the responses might not give an accurate reflection of all group members' views.

From the evaluation results, we conclude that the Talkoo design is perceived as motivating for high school and university level students in that it allows them to successfully build satisfying projects. However, the results concerning the perceived collaboration, which is considered as an essential aspect of project-based learning, illustrate a great difference for high school students compared to university students. As the results show, high school students do not perceive the tool as conducive to collaboration.

As collaboration and teamwork are central parts of physical computing activities in school contexts, they are essential for physical computing kits' successful adoption. Overall, the trial results indicate that the kit (and PELARS workstation) alone cannot sufficiently support collaboration in high school classrooms. Hence, the creation of physical computing toolkits that work efficiently and effectively technology-wise is only one part of the barrier to their adoption into classrooms by practitioners. The results of the user trials highlight the need to provide improved framing for the overall learning environment for classroom use with high school students, as also suggested by the existing literature presented in Section 2. This finding prompted us to expand our research towards asking what kind of additional support teenage users might require for effective collaborative learning and enable us to frame the implementation of Talkoo in a high school classroom context. Drawing from the literature of learning sciences, in the next section, we provide key suggestions regarding the effective implementation of the Talkoo kit for collaborative learning. As discussed in Sections 2.3 and 2.4, these suggestions for effective implementation also have the potential to increase the adoption of physical computing tools in classrooms.

5.2 Implementation of the Talkoo kit for Effective Collaborative Learning

In learning with physical computing, while it is relatively straightforward to identify examples of the new activities and toolkits (as in Section 2.1), it is much harder to arrive at a framework that can support the scaled adoption of these activities and toolkits (as introduced in Section 2.3). One reason for this challenge is that scaling up brings a range of complications that arise from the multiple dimensions of the complex dynamics within learning contexts in terms of the environment, the people with whom learners interact, the toolkits available to support learners and the nature of the knowledge and skills to be learnt [36], which include the following examples:

- space level factors such as layout, furniture, and seating arrangements;
- group characteristics such as size, number and composition;
- group support and facilitation of group processes from adults and/or software systems; and
- group tasks, lessons, curricula and the way they are used to support physical computing activities.

Physical computing activities and toolkits require appropriate pedagogical support to be effective in educational settings. The existence of these activities and toolkits alone does not guarantee that students will be able to achieve various learning objectives after interacting with these activities and kits. Based on the evaluation results, we focus on developing learning activities that will help teachers apply physical computing activities that are pedagogically appropriate to improve students' social and cognitive skills of learning collaboratively.

Based on the existing available evidence from learning sciences, we make four key suggestions for effective implementation of practice-based learning including physical computing implementations [19]

1. Stimulate and encourage effective group work and student discussion during physical computing activities [37][38].
 - Group members must be positively interdependent.
 - Group members must engage in promotive interaction and show a willingness to support each other in their joint efforts to complete the task and achieve the goal.
 - Group members must be individually accountable.
 - Interpersonal and group skills need to be developed.
 - Groups should participate in group processing reflecting on the quality of their group work.
2. Use teacher templates and student worksheets to help plan the stages of physical computing activities and use prompting questions to guide student thinking [39].
3. Use novel and real-life contexts to increase students' situational interests [40].
4. Use integrated illustrations (texts, pictures and multimedia) in physical computing activities [41][42][43]

Table 4 summarises these key suggestions and explains how to implement them in practice. These recommendations and activities are most appropriate for upper secondary and high school students (12–16 year olds) because they are mainly based on findings from studies sampling students of these ages.

Key Suggestions	How they are met?
Group members must be positively interdependent.	Students may be instructed to reach a shared strategy before starting the design and prototype.
Group members must engage in promotive interaction and show a willingness to support each other in their joint efforts to complete the task and achieve the goal.	Students may be asked to follow ground rules for discussions, which aim to provide promotive student interaction.
Group members must be individually accountable.	Students may be asked to share the workload on an equal basis and each student's contribution is needed for the success of the group design.
Interpersonal and group skills need to be developed.	Students' social skills including self-explanation, communication and questioning skills should be targeted.
Groups should participate in group processing reflecting on the quality of their	Students should be provided with enough time, space and guiding questions for their

group work.

Use teacher templates and student worksheets to help plan the stages of a practice-based learning and use prompting questions to guide student thinking.

Use novel and real-life contexts to increase students' situational interests.

Use integrated illustrations (texts, pictures and multimedia) in practice-based learning activities.

purposeful reflection after the activity.

The teacher may use student worksheets to establish ground rules about student discussions and provide guiding questions to promote and lead student thinking and reflection towards the intended learning outcomes.

Interaction design idea narratives can be used to provide a novel and real-life context for students.

Students should be provided multimedia and pictures of potential design and prototypes of the interaction design ideas.

Table 4: Eight key suggestions with implications.

Appendix A provides an example learning activity that puts these recommendations into practice and stands as a best-practice example for the Talkoo toolkit. The example is inspired by the user trial activities. Interventions for strengthening collaborative actions and reflection have been added based on the key recommendations.

6 Limitations of the Study

We designed a toolkit for physical computing using a novel approach to facilitate the building and programming process and encourage collaboration, which proved to motivate students across different educational contexts to create satisfying projects. The evaluation results highlight the necessity to provide key recommendations for implementing the toolkit to guide facilitators in a high school context and better support collaborative learning.

Our integrated approach of a toolkit with key recommendations constitutes a profound starting point to empower young learners and teachers in physical computing that acknowledges not only the issues of technology design but also the dynamics of implementing practice-based learning with programmable toolkits, with a focus on providing more collaborative support and framing to young learners. The evaluation results presented indicate a need for "stronger" pedagogy when implementing physical computing technology with younger learners compared to university students. However, a limitation of our study is that Talkoo was evaluated as part of the larger PELARS system, including the table workstation. This setup might have influenced the actions and perceptions of the participants related to both motivation and collaboration.

Similar to Kafai [9], we consider computational thinking and programming activities social practices that provide a context for making design products important for communities, in which collaboration is an eminent feature. Our viewpoint resonates with that of Wing [10], since we assert that computational thinking is not limited to being able to program a computer but it also involves solving problems and understanding human behaviours in the context of computing. However, as discussed in the results section, low levels of perceived collaboration among young people were observed in this study, specifically for high school students during their practice-based computing activities. Possible reasons for this result are that computational participation might occur differently among different ages, and the contexts of different educational levels might offer various opportunities for collaboration. It might also be the case that a better group ethos among college students, who have been working in groups as part of other modules, helps them to collaborate more fruitfully. Nevertheless, what is clear to us is that collaboration is not likely to occur naturally when

young people are left on their own. If we were to stimulate collaboration as an essential part of computational thinking to shift its conceptualisation towards computational participation, we need to purposefully shape young people's computing activities and their contexts. The recommendations provided in this study might thus be an initial step in this direction.

The Talkoo toolkit acknowledges the demotivating shortcoming of current physical computing kits because it leverages common obstacles of circuiting, programming and debugging and provides more space and time for collaborative actions. The key recommendations provided acknowledge the evaluation insights that the kit has potential to enable teenage children to successfully create satisfying projects within a short time. However, these suggestions should be investigated in-depth to judge whether implementing the key suggestions with Talkoo will have the desired effect on collaborative learning. This question will shape our immediate future work.

7 Conclusion

We presented a novel plug-and-play toolkit in a visual programming environment accompanied by eight key suggestions for a successful implementation by high school teachers, with a view to improving students' motivation and collaboration skills in project-based physical computing activities. The findings show that, for young teenage learners, we need to not only think about kit design but also of *learning contexts* in which these kits are implemented. Physical computing kits, ideally, should include appropriate guidance for their implementation in educational contexts. Overall, our work expands the field of programming with physical computing for teenage children with a highlight on empowering both teachers and students. The findings can be beneficial not only for educators but also for researchers and designers of physical computing learning environments.

8 Acknowledgements

The PELARS project received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 619738.

9 References

- [1] S. L. Martinez, and G. S. Stager, *Invent to Learn: Making, Tinkering, and Engineering in the Classroom*, Torrance, CA: Constructing Modern Knowledge Press, 2013.
- [2] P. Blikstein, "Computationally enhanced toolkits for children: Historical review and a framework for future design," *Foundations and Trends® in Human-Computer Interaction*, vol. 9, no. 1, pp. 1–68, 2015. <https://doi.org/10.1561/1100000057>
- [3] P. Mulholland, S. Anastopoulou, A. Collins, M. Feisst, L. Kerawalla, and M. Wright, "nQuire: Technological support for personal inquiry learning," *IEEE Trans. Learn. Technol.*, vol. 5, no. 2, pp. 157–169, 2012.
- [4] M. Scardamalia, "Getting real about 21st century education," *J. Educational Change*, vol. 2, pp. 171–176, 2001.
- [5] M. Cukurova, P. Hanley and A. Lewis, "An international rapid evidence assessment (REA) of practical science policy and practice", Report for the Gatsby Foundation: London, UK, 2017.
- [6] P. Blatchford, P. Kutnick, and E. Baines, "Pupil grouping for learning in classrooms: Results from the UK SPRinG Study," Paper presented at symposium 'International Perspectives on Effective Groupwork: Theory, Evidence and Implications', American Educational Research Annual Meeting, Chicago, April 2007.
- [7] P. Blatchford, P. Bassett, and P. Brown, "Teachers' and pupils' behaviour in large and small classes: A systematic observation study of pupils aged 10/11 years," *J. Educational Psychology*, vol. 97(3), pp. 454–467, 2005.
- [8] P. Blatchford, E. Baines, C. Rubie-Davies, P. Bassett, and A. Chowne, "The effect of a new approach to group-work on pupil-pupil and teacher-pupil interaction," *J. Educational Psychology*, vol. 98(4), pp. 750-765, 2006.
- [9] Y. B. Kafai, "From computational thinking to computational participation in K-12 education," *Communications of the ACM*, vol. 59, no. 8, pp. 26–27, 2016. Available: <https://doi.org/10.1145/2955114>
- [10] J. M. Wing, "Computational Thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006. Available: <https://doi.org/10.1145/1118178.1118215>

- [11] M. Przybylla and R. Romeike, "Physical computing and its scope-towards a constructionist computer science curriculum with physical computing," *Informatics in Education*, vol. 13, no. 2, p. 225, 2014.
- [12] J. Chan, T. Pondicherry, and P. Blikstein, "LightUp: An augmented, learning platform for electronics," in *Proc. 12th Int. Conf. Interaction Design and Children*, New York, NY, 2013, pp. 491–494. Available: <https://doi.org/10.1145/2485760.2485812>
- [13] OECD. "Education at a glance 2014," Paris: Organisation for Economic Co-operation and Development. Available: <http://www.oecd-ilibrary.org/content/book/eag-2014-en>
- [14] A. Bdeir and P. Rothman, "Electronics as material: LittleBits," in *Proc. 6th Int. Conf. Tangible, Embedded and Embodied Interaction*, New York, NY, 2012, pp. 371–374. Available: <https://doi.org/10.1145/2148131.2148220>
- [15] J. Sadler, K. Durfee, L. Shluzas, and P. Blikstein, "Bloctopus: A novice modular sensor system for playful prototyping," in *Proc. 9th Int. Conf. Tangible, Embedded, and Embodied Interaction*, New York, NY, 2015, pp. 347–354. Available: <https://doi.org/10.1145/2677199.2680581>
- [16] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, Y. Kafai, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009. Available: <https://doi.org/10.1145/1592761.1592779>
- [17] T. Booth and S. Stumpf, "End-User Experiences of Visual and Textual Programming Environments for Arduino," in *End-User Development*, Y. Dittrich, M. Burnett, A. Mørch, and D. Redmiles, Eds., Heidelberg: Springer Berlin, 2013, pp. 25–39. Available: https://doi.org/10.1007/978-3-642-38706-7_4
- [18] R. Luckin, B. Blich, A. Manches, S. Ainsworth, C. Crook, and R. Noss, *Decoding Learning: The Proof, Promise and Potential of Digital Education*, London: NESTA, 2012.
- [19] P. Davies, G. Kent, D. Laurillard, M. Mavrikis, R. Noss, D. Pratt, S. Price, *The Impact of Technological Change on STEM Education*. London, UK: Royal Society Vision Project, 2013.
- [20] M. E. Webb, "Affordances of ICT in science learning: Implications for an integrated pedagogy," *Int. J. Science Education*, vol. 27, no. 6, pp. 705–735, 2005. doi:10.1080/09500690500038520.
- [21] A. S. Yeung, P. G. Taylor, C. Hui, A. C. Lam-Chiang, E.-L. Low, "Mandatory use of technology in teaching: Who cares and so what?" *British J. Educational Technol.*, vol. 43, no. 6, pp. 859–870, 2012. doi:10.1111/j.1467-8535.2011.01253.x.
- [22] A. Clark-Wilson, C. Hoyles, R. Noss, P. Vahey and J. Roschelle, "Scaling a technology-based innovation: Windows on the evolution of mathematics teachers' practices," *ZDM Mathematics Education*, vol. 47, no. 1, pp. 79–92, 2015. doi:10.1007/s11858-014-0635-6.
- [23] S. Hegedus and J. Roschelle, *The SimCalc Vision and Contributions*. Netherlands: Springer, 2013.
- [24] C. Coburn, "Rethinking scale: Moving beyond numbers to deep and lasting change," *Educational Researcher*, vol. 32, no. 6, pp. 3–12, 2003.
- [25] D. Hung, K. Lim, and D. Huang, "Extending and Scaling Technology-Based Innovations Through Research: The Case of Singapore," in *Inspired by Technology, Driven by Pedagogy: A Systemic Approach to Technology-Based School Innovations*, OECD Publishing, 2010, pp. 89–102.
- [26] T. Neutens and F. Wyffels, "Teacher professional development through a physical computing workshop," in *Proc. 11th Workshop in Primary and Secondary Computing Education*, Munster, Germany, p.108-109 2016.
- [27] D. Spikol, K. Avramides, E.-S. Katterfeldt, E. Ruffaldi, and D. Cuartielles, "Opportunities with digital fabrication through learning analytics," in *Exploring the Material Conditions of Learning: The Computer Supported Collaborative Learning (CSCL) Conf. 2015*, Gothenburg, Sweden, vol. 2, pp. 697–698. Available: <http://www.isls.org/cscl2015/papers/CSCL2015ProceedingsVolume2.pdf>
- [28] E. Ruffaldi, G. Dabisias, L. Landolfi, and D. Spikol, "Data collection and processing for a multimodal learning analytic system," in *IEEE SAI Conference*, London, UK, 2016.
- [29] M. Cukurova, R. Luckin, E. Millán, and M. Mavrikis, "The NISPI framework: Analysing collaborative problem-solving from students' physical interactions," *Computers & Education*, vol. 116, supplement C, pp. 93–109, 2018. <https://doi.org/10.1016/j.compedu.2017.08.007>
- [30] Y. Mor and N. Winters, "Design approaches in technology-enhanced learning," *Interactive Learning Environments*, vol. 15, no. 1, pp. 61–75, 2007. <https://doi.org/10.1080/10494820601044236>
- [31] E. Björgvinsson, P. Ehn, and P.-A. Hillgren, "Participatory design and "democratizing innovation," in *Proc. 11th Biennial Participatory Design Conf. PDC '10*, New York, NY, 2010, p. 41. Available: <https://doi.org/10.1145/1900441.1900448>
- [32] E.-S. Katterfeldt, D. Cuartielles, D. Spikol, N. Ehrenberg, „Talkoo: A New Paradigm for Physical Computing at School," in: Proceedings of the The 15th International Conference on Interaction Design and Children, IDC '16. ACM, New York, NY, USA, pp. 512–517, 2016. <https://doi.org/10.1145/2930674.2935990>
- [33] D. Spikol, N. Ehrenberg, V. Bahtijar, D. Cuartielles, and N. Valkanova, "Designing a Visual Programming Platform for Prototyping with Electronics for Collaborative Learning," in A.-M. Nortvig, B. H. Sørensen, M. Misfeldt, R. Ørngreen, and B. Allsopp (Eds.), *Designs for Learning*, Aalborg: Aalborg University, 2016, pp. 102–109. Available: http://www.designsforlearning2016.aau.dk/digitalAssets/196/196362_dfl_short-papers-samlet_final-pdf.pdf#page=102
- [34] E. L. Deci and R. M. Ryan, *Self-determination*. John Wiley & Sons, 2010.
- [35] M. Hoegl and H. G. Gemuenden, "Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence," *Organization Science*, vol. 12, no. 4, pp. 435–449, 2001.

- [36] R. Luckin, *Re-designing Learning Contexts: Technology-Rich, Learner-Centred Ecologies*. London: Routledge, 2010.
- [37] R. Taraban, C. Box, R. Myers, R. Pollard, and C. W. Bowen, "Effects of active-learning experiences on achievement, attitudes, and behaviors in high school biology," *J. Res. Science Teaching*, vol. 44, no. 7, pp. 960–979, 2007.
- [38] M. P. Freedman, "Relationship among laboratory instruction, attitude towards science, and achievement in science knowledge," *J. Res. Science Teaching*, vol. 34, no. 4, pp. 343–357, 1997.
- [39] C. W. Keys, B. Hand, V. Prain, and S. Collins, "Using the science writing heuristic as a tool for learning from laboratory investigations in secondary science," *J. Res. Science Teaching*, vol. 36, pp. 1065–1084, 1999.
- [40] D. H. Palmer, "Students' interest generated during an inquiry skills lesson," *J. Res. Science Teaching*, vol. 46, no. 2, 2009, pp. 147–165.
- [41] C. Y. Haslam and R. J. Hamilton, "Investigating the use of integrated instructions to reduce the cognitive load associated with doing practical work in secondary school science," *Int. J. Science Education*, vol. 32, no. 13, pp. 1715–1737, 2010.
- [42] N. Marcus, M. Cooper, J. Sweller, "Understanding instructions," *J. Educational Psychology*, vol. 89, no. 1, pp. 49–63, 1996.
- [43] R. E. Mayer and R. Moreno, "A split-attention effect in multimedia learning: Evidence for dual processing systems in working memory," *J. Educational Psychology*, vol. 90, no. 2, pp. 312–320, 1998.

Appendix A

An example activity: How can we help kids get over their fear of the dark?

Intended Learning Outcomes:

- 1) Students learn visual programming modules and functions of Talkoo together.
- 2) Students learn to design and create a prototype of an idea from scratch working together as a group.
- 3) Students learn to problem solve collaboratively.

Activity time: 120–180 mins

Number of students: Small groups of 3–6 students

Activity Steps:

- 1) The teacher introduces the background narrative. Students are instructed to design interactive companions. These are objects with individual personalities that are afraid of the dark but are part of an exploratory adventure that children can take through the room. The aim is to create objects that dim as children become more comfortable without them at night and the students are asked to use the PELARS Talkoo kit to prototype this idea.
- 2) The teacher introduces the learning outcomes to the students and emphasises the value of thinking together.
- 3) The teacher introduces a list of ground rules about talking and working together (see table A1). These rules are printed on the student sheets and given to the students to follow during their practice-based learning activities.
- 4) Each group is asked to agree on a strategy about how design and create a prototype for the idea introduced to them in step 1 using the Talkoo kit. For example, they might use a potentiometer and light sensor connected with an LED through the

programming logical block of “Map.” In this case, they might adjust the potentiometer to show the length of time that has passed, and the LED reacts to the amount of darkness around it.

- 5) Students present their solutions to another group and discuss the plausibility of each group’s idea and its pros and cons. Students are reminded to follow the ground rules for their discussions.
- 6) Students are asked to revise their final strategy for the design and prototyping of the idea after receiving feedback from the other group.

Ground Rules for Student Discussion

Listen to and think about others’ ideas and do not talk over anyone.

Do not take design and prototyping actions unless you reach a shared agreement.

Ask for reasons and explanations for every idea suggested before taking design and prototyping actions.

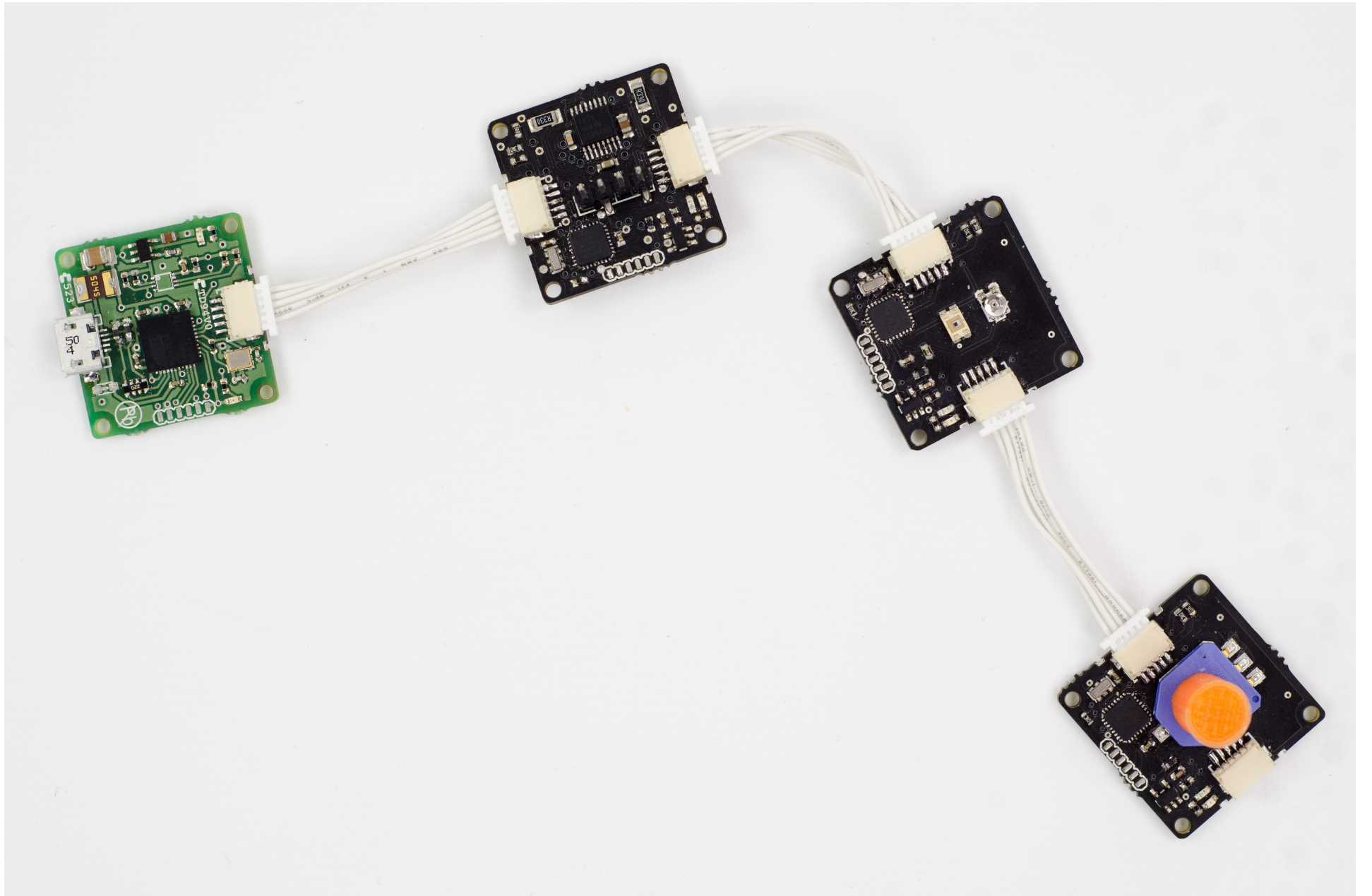
If anyone finds it hard to join in and stays silent for a long time, encourage them to share their opinion.

Make group decisions that you all agree with.

Ask each other questions to help everyone understand each participant’s ideas.

Table A1: Ground Rules for Student Discussion

- 7) Students are asked to share the workload to complete the design and prototyping of their final strategy as fairly as possible depending on the skills, knowledge and interests of the students.
- 8) The group brings their idea into life and seeks help from each other whenever they need while they are working on their actions.
- 9) Students are asked to reflect on their practice. They are asked to provide answers to three basic questions:
 - (A) What would they keep exactly the same the next time?
 - (B) What would they change completely the next time?
 - (C) What would they keep but adapt and improve the next time?
- 10) They present their final product to the teacher who leads the final discussion with the following guiding questions:
 - What did the group agree to build and program, and why?
 - What were the most challenging parts of each piece of work (shaping, programming, plugging, cutting etc.) shared among the participants?
 - How did the group explain what they intended to create to the other group?
 - Which ground rules were useful and which were not?
 - What is the most challenging part of this activity, and why?
 - What would you do differently next time to improve the whole process?





Phases	Focus	Context	Students	Approach	Instruments	Users
Field Testing	Usability issues of Talkoo	Ars Electronica Festival	High school	Exploratory: observations, workshops, <i>data from PELARS on kit usage</i>	Workshop, participatory-design sessions, <i>data analytics</i>	14 in workshops 86 ad-hoc
User Trials	Collaboration, satisfaction with outcomes and motivation	Classrooms at formal institutions	High school and university	Mixed methods	Pre, in-situ, <i>post surveys</i> , data analytics, structured observations	57 university 20 high school in 44 group sessions

Table 1. Design phases, contexts and general approach. Approach and instruments reported in this article are set in italics.

University		High School	
N=	37	N=	7
Mean	3.92	Mean	4.25
SD	1.01	SD	0.89

Table 2. Students' satisfaction with their projects (group based).

All Students		Engineering		Design		High School	
N=	77	N=	19	N=	38	N=	20
Mean	3.36	Mean	4.32	Mean	3.79	Mean	1.65
SD	1.29	SD	0.58	SD	0.81	SD	0.81

Table 3. Perception of collaboration when using the PELARS system (table and toolkits; individual based).

Key Suggestions

Group members must be positively interdependent.

Group members must engage in promotive interaction and show a willingness to support each other in their joint efforts to complete the task and achieve the goal.

Group members must be individually accountable.

Interpersonal and group skills need to be developed.

Groups should participate in group processing reflecting on the quality of their group work.

Use teacher templates and student worksheets to help plan the stages of a practice-based learning and use prompting questions to guide student thinking.

Use novel and real-life contexts to increase students' situational interests.

Use integrated illustrations (texts, pictures and multimedia) in practice-based learning activities.

How they are met?

Students may be instructed to reach a shared strategy before starting the design and prototype.

Students may be asked to follow ground rules for discussions, which aim to provide promotive student interaction.

Students may be asked to share the workload on an equal basis and each student's contribution is needed for the success of the group design.

Students' social skills including self-explanation, communication and questioning skills should be targeted.

Students should be provided with enough time, space and guiding questions for their purposeful reflection after the activity.

The teacher may use student worksheets to establish ground rules about student discussions and provide guiding questions to promote and lead student thinking and reflection towards the intended learning outcomes.

Interaction design idea narratives can be used to provide a novel and real-life context for students.

Students should be provided multimedia and pictures of potential design and prototypes of the interaction design ideas.

Table 4: Eight key suggestions with implications.