

# **Architectural Authorship in Generative Design**

**Athina Theodoropoulou**

September 2007

This dissertation is submitted in partial fulfilment of the requirements for the degree of Master of Science in Adaptive Architecture & Computation from the University of London

**The Bartlett School of Graduate Studies  
University College London**

## Abstract

---

The emergence of evolutionary digital design methods, relying on the creative generation of novel forms, has transformed the design process altogether and consequently the role of the architect. These methods are more than the means to aid and enhance the design process or to perfect the representation of finite architectural projects. The architectural design philosophy is gradually transcending to a hybrid of art, engineering, computer programming and biology. Within this framework, the emergence of designs relies on the architect-machine interaction and the authorship that each of the two shares.

This work aims to explore the changes within the design process and to define the authorial control of a new breed of architects-programmers and architects-users on architecture and its design representation. For the investigation of these problems, this thesis is to be based on an experiment conducted by the author in order to test the interaction of architects with different digital design methods and their authorial control over the final product. Eventually, the results will be compared and evaluated in relation to the theoretic views. Ultimately, the architect will establish his authorial role.

### Keywords

Authorship, authorial role, design process, emergent, computation, self organizing.

**Word Count**

10.279

## Acknowledgements

---

I would like to thank my supervisors, Chiron Mottram for his encouragement and support and Karen Martin for her guidance, useful comments and positive thoughts throughout the project. Many thanks to Ava J.Fatah for her interest and valuable advice in the initial stages of this project.

Especially, I would like to thank Christian Derix for offering his "Faulty Towers" plug-in to AutoCAD, for the realization of the experiment, as well as his valuable remarks.

Finally, special thanks to all who participated in my experiment: Charis Athanasopoulou, Rafail Bakolas, Charis Charalampous, Assawin Choochottavorn, Shih-Ying-Chung Chou, Chris Leung, Adam Doulgerakis, Sonam Gupta, Martin Kaftan, Tassos Kanellos, Veroloucy Kasserli, Maria Kerkidou, Athanasios Kolios, Anna Laskari, Simin Li, Antonia Mauroudi and Alexandros Tsolakis.

# Contents

---

<i>Abstract</i>	2
<i>Acknowledgements</i>	3
<i>Contents</i>	4
<i>List of figures</i>	5
<i>Introduction</i>	6
Thesis Overview	8
<b><i>PART I: Background and literature review</i></b>	<b>10</b>
1. The Design Process	11
2. Generative Design	14
3. Authorship	16
<b><i>PART II: The experiment</i></b>	<b>20</b>
1. Methodology:	21
1.1 The 'designers'	21
1.2 The programs	21
1.3 The concept of the experiment	24
2. Case Studies:	25
2.1 Designing with CAD	26
2.1.1 The designs	26
2.1.2 The designers	28
2.1.3 The program	29
2.2 Designing with Programming	30
2.2.1 The designs	30
2.2.2 The designers	32
2.2.3 The program	34
2.3 Designing with a Self-Organizing system	35
2.3.1 The designs	35
2.3.2 The designers	38
2.3.3 The program	39
2.4 The 'digital' design process: General Observations and Comments	40
2.5 Authorship	43
<i>Discussion</i>	48
The Architect-user	48
The Architect- programmer	52
Further thoughts	54
<i>Conclusion</i>	55
<i>References</i>	57
<i>Appendix I: Participants' Comments</i>	60
<i>Appendix II: Case Study results</i>	65

## List of figures

<b>Fig. 1</b> <i>Tower of Pisa</i> .....	22
< <a href="http://architecture.about.com/library/blleaningtowerofpisa.htm">http://architecture.about.com/library/blleaningtowerofpisa.htm</a> > [September 2007]	
<b>Fig. 2</b> <i>Eiffel Tower</i> .....	22
< <a href="http://www.todayinsci.com/3/3_31.htm">http://www.todayinsci.com/3/3_31.htm</a> > [September 2007]	
<b>Fig. 3</b> <i>The Gherkin</i> .....	22
< <a href="http://pgoh13.free.fr/london_inphotos.php">http://pgoh13.free.fr/london_inphotos.php</a> > [September 2007]	
<b>Fig. 4</b> <i>CAD designs, group 1.</i> .....	27
by participants of the experiment	
<b>Fig. 5</b> <i>CAD designs, group 2.</i> .....	27
by participants of the experiment	
<b>Fig. 6</b> <i>Programming designs, group 1.</i> .....	31
by participants of the experiment	
<b>Fig. 7</b> <i>Programming designs, group 2.</i> .....	32
by participants of the experiment	
<b>Fig. 8</b> <i>'Faulty Towers', group 1.</i> .....	36
by participants of the experiment	
<b>Fig. 9</b> <i>'Faulty Towers', group 2.</i> .....	37
by participants of the experiment	
<b>Fig. 10</b> <i>Projects with one concept throughout the experiment.</i> .....	42
by participants of the experiment	
<b>Fig. 11</b> <i>Representational graph showing the designers' views on the computational methods used at the experiment.</i> .....	43
by author	
<b>Fig. 12</b> <i>Graphic representation of the participants' answers about what an architect 'author'.</i> .....	45
by author	
<b>Fig. 13</b> <i>Graphic representation of the participants' views on the authorial control that the software, its designer and user have over the final outcome.</i> .....	46
by author	
<b>Fig. 14</b> <i>Graphic representation of the participants' view on the influence of the use of 'self organizing' systems in the architectural process.</i> .....	47
by author	

## Introduction

---

With the emergence of Information Technology, advanced computational power and knowledge have taken architecture into new fields of design exploration and experimentation. This thesis aims to explore the authorial control of the architect over the generative designs produced by evolutionary computations and seeks the role of the user and the designer behind such programs.

The new design methods that are being developed do not confine themselves to the detailed representation of finite ideas and existing designs, applied more easily at the final stages of the design process. The growing trend in computer tools is the emergence of creative evolutionary design methods that are capable of more than the replication and parameterization of complex systems but share generative and simulation attributes that 'give birth' to those very systems. Creative evolutionary design tools are implementing the state-of-the-art computing mechanisms to actually give life to 'artificial creativity'.

An autonomous system, however, brings to the surface issues of control and human-like behavior producing modules that threaten with extinction the very human nature that has invented them. In architecture, in particular, the role of the architect seems to be excluded of the creative process while the design is self-made. The incorporation of emergent systems into a general software architectural methodology, in a way, suspends the control that the architect has over the

final outcome and sets questions like the one Nigel Cross pursued: "Can a machine design?" only to transform it after years of research to the corollary question "How do people design?" [Cross, 2001] The computational attempts to simulate both nature and humans only aim to produce a design that will be to the satisfaction of the designer. In addition to that, the architect is upgraded to a software designer, thus, regaining the control of these processes through a different role. Rolf Hughes in his treatise on "The Semi-living Author" remarks: "Whether this represents an abdication of authorial intention or its ultimate expression (a Promethean attempt to seize the original rights of the 'auctor vitae', perhaps) remains an open question". [Anstey, Grillner, Hughes, 2007, p.132] In other words, maybe it is not the human nature that is threatened- by computational studies on the implementation of human properties and processes into computer software- but merely, the next step of architectural evolution that is about to come, embracing within its structure the technological progress.

Although the world refers to The Gherkin (**fig.3**) as Norman Foster's Swiss Re Tower, there is a whole team behind the designs and effectuation of this project, a multi- authored nature of the work by Foster and Partners that is undoubtedly strongly supported by the most advanced computation design techniques. However, the architectural persona behind the conceptual creation of the actual building remains indisputably "the world's most famous architect". "Whether Foster could ever become as faceless as Fosters- remains to be seen." [Anstey, Grillner, Hughes, 2007, pp.12-13]

## Thesis Overview

The research question that this thesis will attempt to answer through its development is: In what way the possibility of generative systems affects the design process and its final outcome? And through the investigation of the evolution of the design process: In what ways the authorial role of the architect-user and the architect-programmer is threatened? Is it really threatened or is it simply evolving?

This thesis will begin with a literature review of the main views concerning the principles that govern the design process as this is seen through philosophy and the research of design education. An effort will be made to explore how design thinking works, taking the cognitive processes of the designer into account. At the next step, the emergence of the evolutionary architecture will be presented in relation to the role of the architect extending into the 'breeder of forms' and the implications that such a prospective rises. Finally, the authorial role of the architect will be revealed, as it is adapted through the evolution of the design values and accompanying aspects of modern design.

The statements presented through the theoretical approach of this paper will set the issue of authorship and architecture under consideration. Consequently, an experiment conducted by the author will be presented aiming to identify in practice the authorial control of the designer through different computational design techniques. The results of this experiment will be revealed and analysed in detail and an effort will be made to draw some conclusions about the designing



behaviour of the participants and their manipulation by the programs.

In the end, the findings of the study will be discussed in accordance to the theoretic views on the topics under consideration at both conceptual and practical levels. Furthermore, this paper will conclude by giving an overall evaluation in regards to the future prospective of architectural design.

## **PART I: Background and literature review**

*"For the poet is a light and winged and holy thing, and there is no invention in him until he has been inspired and is out of his senses, and the mind is no longer in him: when he has not attained to this state, he is powerless and is unable to utter his oracles. [...] for not by art does the poet sing, but by power divine."*

[Plato, ION, 380 BC, 534b3-c1]

Malcolm McCullough in *Digital Ground* references the official guidebook of *The World of Tomorrow* to highlight that *"the true poets of the 20<sup>th</sup> century are the designers, the architects and the engineers who glimpse some inner vision, create some beautiful figment of the imagination, and then translate it into valid actuality for the world to enjoy"*. [McCullough, 2004, p.12] Architectural design is much more than an artistic expression. It involves a combination of rational and imaginative thinking modifying the environment and its social parameters.

In this part, a literature based analysis will be made in order to call to attention the basic characteristics of the traditional design process as it is established through years of practice and educational experience. New approaches to architecture practice based on computer technologies that have been developed during the last decades will be presented in order to exhibit the changes that digital drawing has brought to the design process and the challenges that the role of the

architect in 'reflective practice' faces, as the potentiality of emergent architecture is gradually becoming the new reality in the architectural field. Finally, the notion of 'authorship' will be introduced and a historical analysis of the authorial control that the architect has over the building and its design representation will be made, as an ominous new era for the role of the architect is dawning.

## 1. The Design Process

---

The complexity of human mental processes is indisputable. The human brain, relying on experience, retrieving knowledge from memory and forming associations between seemingly unrelated items, exceeds any effort of 'artificial intelligence' ever made. The reason for that is simple: scientists and cognitive theories even today fail to fully understand the course of cognitive growth. [Neisser, 1967] Intuition, perception and imagination drive the creative processes inside one's mind and comprise the human skills that feed, in turn, architectural creativity. In order to analyse the design process and relate it with the use of modern media, it is necessary to reveal, first, its associations with these cognitive processes of the brain.

*"One of the most difficult aspects of understanding designing has always been that too many divergent acts occur simultaneously, defying simple description."* [Habraken, Gross, 1988, p.151] Because of this fact, designing is a complex process and there cannot be a defined sequence of actions or even identifiable

separable actions that the designer follows to succeed his task. Instead, a good process of design is a reflective process in which the designer is engaging in a conversation with the situation as it unfolds. He confronts the consequences of his actions and needs to evaluate the implications before proceeding with the next move. What is interesting about this process- which Schön calls 'reflection-in-action'- is the ability of the designer to recognise the changes and redirect his strategy according to the situation's 'back-talk' thus generating more ideas. [Schön, 1991]

In addition to that, the designer does not approach each design with a *tabula rasa*. He perceives each situation both as unique and as familiar. The literature background and his personal experience are valuable tools that he carries even when designing the simplest objects. The load of this knowledge is both an asset and a drawback. On the one hand, it may be harder than one imagines, to break free of these pre-structures to create a totally original design solution, even when wishing and encouraged to do so. Schön, on the other hand, saw this as a key aspect of reflection thought [Schön, 1991, pp.138-150], while Mitchell noted, in "A new agenda for computer-aided design", that it is the knowledge of the work of others as well as the criticism on one's own work that leads him to constant stylistic evolution; which long-term produces a stylistic evolution to the framework of architecture. [McCullough, Mitchell, Purcell, 1990, p.6] "*Design is therefore both the transmission and transformation of pre-structures, a process of elaboration and discovery, within which every solution may be unique*" [Hillier, Leaman, 1974, p.5]

Designers have the ability to recognise emergent sub shapes within the structure of a design and at the same time their mental connotations and reasoning of them gives a different interpretation on different occasions. Human visual perception, object recognition, scene interpretation, linking vision with action and reasoning, are all characteristics that evolutionary computation struggles to achieve in simulation models; but human vision is unlike computer scanning of a digital model or processing an image recognition algorithm. Neisser, in his book *Cognitive Psychology* reports that humans have these abilities inscribed in their DNA and RNA, as experimental findings in biochemistry have hinted, nevertheless, this is not what counts for cognitive psychology. He points out that it is not the "hardware" that is important but, similarly to programming, the program that it runs. However, contrary to any promising computational system, the human brain is selective- both in memory storage and processing- and is in no way neutral or passive to the information it receives. [Neisser, 1967, pp.3-11]

*"It seems to me that an understanding of the complex context in which people operate must eventually enrich our understanding of who the individual is and what the individual does. But to do so we need to abandon the Ptolemaic view of creativity, in which the person is at the centre of everything, for a more Copernican model in which the person is part of a system of mutual influences and information"* [Sternberg, 1988, p.336]

The interplay between designer and design, practitioner and practice, suggests a form of dialogue relating explicit and implicit qualities of the human

nature, thus, creating a unique equilibrium that comprises the design process. Creativity, nevertheless, is a phenomenon that results within a social and historical milieu. Therefore, is cognition enough for the evolution of the design process?

## 2. Generative Design

---

"By this art you may contemplate the variation of the 23 letters..."

[Borges, 1970, p.78]

The development of design computing brought about considerable investigation and debate. For the last 50 years the design process, and in particular the architectural design process, has been established as a whole new research field.

Since Ivan Sutherland's pioneering Sketchpad system -based on the idea of structured design representation in computer memory- was built, several issues concerning architectural theory and education were raised. The very structure of computer drawings caused ambiguous interpretations and brought to the surface the architect-machine dialogue that Negroponte already attempts to describe in 1970. [Negroponte, 1970, pp.83-87] Nowadays, digital computing is being broadly popularised by CAD software tools. The 2-D graphic platforms have, after a long period of trial and mistrust, earned a place in every architectural office while their extension into 3-D object modelling tools is gaining ground fast. In the

twentieth century, though, the application of Darwinian evolution methods to software development is attempted and a new cycle of debates begins, concerning the newest media platforms.

In *On Growth and Form* D'Arcy Thompson [1961] suggests that the form of living things is not predetermined genetically, but is a result of precise mathematical principles and the physical environment. Evolutionary computing based on the principles of evolutionary biology and mimicking the genesis of living organisms, seeks to create the generative process for architectural form as well. Designers- programmers describe the rules in a genetic language which produces the code- script of instructions for form- generation. This method is used to identify the underlying design rules that build difficult geometries translating conceptual designs into structural descriptions and simulate the development of prototypical forms which can afterwards be evaluated based on their performance in a simulated environment. [Branko, 2003, pp.24-27] In other words, the architects design the, more or less, autonomous systems which develop, evolve, or design architectural structures, objects, or spaces without any kind of blueprint. John Frazer has defined emergent architecture as "*a property of the process of organizing matter rather than a property of the matter thus organised*" - it is a "*process driven architecture*" which emerges "*on the very edge of chaos, where all living things emerge, and it will inevitably share some characteristics of primitive life forms*". [Frazer, 1995, p.103]

"Architects wishing to use this new tool must not only become hackers (so that they can create the code needed to bring extensive and intensive aspects together) but also be able "to hack" biology, thermodynamics, mathematics, and other areas of science to tap into the necessary resources." [DeLanda, 2002, p.6]

Manuel DeLanda in *Deleuze and the Use of the Genetic Algorithm in Architecture* only describes what John Frazer in 1995 had called the "extended architect", a new generation of architects that will comprise the 'genetic pool of ideas'. [Frazer, 1995, p.100] Only, DeLanda will wonder: "Will these future authors be satisfied with the role of breeders of virtual forms?" [DeLanda, 2002, p.4]

### 3. Authorship

---

What does it mean for an architect- designer to be an author and what is he authoring? The issue of authorship has been an area of intense scrutiny over the last forty years. Contemporary practices of architecture have produced a hybrid of designer- programmer that focuses on how architecture is produced, through self-generative systems, annulling the architect as an author.

Since the days of Alberti, who identified the architect as a figure 'semi-divine' trying to articulate a 'divine sense of beauty' and established the idea that a meaningful authorial relation can be sought between an architect and a building, architecture has been identified with the persona of the architect. However, Alberti extends this idea so as to detach the architect



from the building as physical object and only attributes to him the authorial control over the representation of the building, which is the idea and principles behind the actual creation, the judgment behind the doing. He claims that *"to make something that appears to be convenient for use, and that can without doubt be afforded and built as projected, is the job not of the architect but of the workman. But to preconceive and to determine in the mind and with judgment something that will be perfect and complete in its every part is the achievement of such a mind as we seek."*[Alberti, 1485, p.315]

Roland Barthes in "the death of the author" in 1968, challenges the notion of authorship altogether. Barthes suggests the emergence of a new authorial persona. He disconnects the creation from the Author and assigns to it autonomous voice in the realm of a multi-dimensional space, where it is in the consciousness of its interpreters to be attributing a meaning. Once the creation comes to light, the author, according to Barthes, *"enters into his own death"*. [Barthes, 1977, p.142]

History of modern architecture, on the other hand, has proven over time that creative and original works of architecture are always identified with the person of the architect. Barthes admits that *"the explanation of a work is always sought in the man or woman who produced it, as if were always in the end, through the more or less transparent allegory of the fiction, the voice of a single person, the author 'confiding' in us"*. [Barthes, 1977, p.143] Tim Anstey will point out that *"the Albertian figure of the architect is one that proves extremely hard for architects, and for architectural*

*history and theory, to escape.*" [Anstey, 2005, p.304]  
*"What remains distinctive in architectural culture is the tendency of almost any kind of discussion to precipitate authors, individuals with names who stand for an intention or set of intentions realised in buildings or articulated through visual media and publication."*  
[Anstey, Grillner, Hughes, 2007, p.12]

However, for the last 50 years the growth of computer technologies and the rise of user-oriented (participatory) design techniques set new facts for the revision of architectural authorship. The evolution of emergent architecture brought about 'nameless' architectures created by self-organising systems and shook up the traditional notions of architectural authorship. Rolf Hughes, in his paper "The semi-living Author: Post-human", mentions that *"once the autopoietic system is up and running, the author would thereafter seem to be not so much dead (or terminally afflicted) as written out of the equation altogether."* [Anstey, Grillner, Hughes, 2007, p.132]

On the other hand, Habraken and Gross [1988, p.155], in their paper "Concept design games", explain that *"the [design] game may be developed as much by the players as by the initial creator of the game"*. If the design process through a computer program can be considered as a design game where the developer-designer of the program has set the stage for the player-designer to work on, then, it can be in parallel concluded that the software designer creates the platform and determines the rules that define the design process through the particular software that the designers engage to. Therefore, the design process in this case may be developed as much by

the designers as by the software designer. The question now is: in what degree the design is authored by the software, the software designer or the software user?

The exploration of how systems, rather than architects, order architecture is wide. The issue here is not to preserve the traditional nodes of authorship but to expose the danger of losing control and self-abdicating for the architect who uncontrollably surrenders to the machine instead of reflecting with it. Understandably, experiments in authorial abdication and control within the realm of self generative design need to re-establish the authorial role of the architect in order to promote a creative equilibrium.

## **PART II: The experiment**

For the needs of this research and in order to evaluate the authorial role of the designer in the realm of generative design, an experiment was conducted assigning a group of people to produce a series of designs, using a variety of computer-based techniques.

The experiment was designed to assess the impact of the computer program, on the design process and its final outcomes, by evaluating the way users perform a simple task. Post-task questionnaires were used to identify the views of the participants on the emergent themes in combination with careful observation of them during the experiment. Both qualitative and quantitative analyses were performed to reach to satisfactory conclusions.

In this experiment four main concerns were to be noted. The cognitive processes followed by the participants during their interaction with the digital media, the control that the participants had over the various computational methods and the final designs, the control of the used tools over the participants and the final designs and, finally, the flexibility of the participants to mixing different digital media. Out of these assessments, the experiment is aiming to reveal the authorship relations within the 'digital' design process.

In the following units, this experiment as well as the concept behind its methodology will be described in detail. The results will be analysed, combined and presented in relation to the participants' views and comments, expressed through the questionnaires. Finally, there will be an overall evaluation of the experiment.

## 1. Methodology:

---

### 1.1 The 'designers'

The experiment involved a limited number of 17 engineers most of which are architects, students of the MSc Adaptive Architecture and Computation at UCL. The profile of the participants is quite homogeneous. The individuals that participated in this research have at least one year of working experience, part of which involved their participation in the designing process of real projects, and several years of experience in student design projects. In addition to that, all of them are efficient users of several CAD programs and have one year's at least programming experience and exposure to the theory and techniques of generative programming. Their programming skill is what differentiates the participants of this experiment, as it is not yet common for most architects to have such knowledge. For the needs of this experiment the participants will be referred to as the 'designers'.

### 1.2 The programs

The designers were required to work individually so as to produce the design of 'a tower' using three different base programs which could, at their judgment, be combined with more computer programs of their choice. The exercise comprised 3 parts, which reflected the three different approaches for the same subject and the designers had 30 minutes for the completion of each design.

At the beginning of the experiment the designers were briefed on the experiment. The need of design results that would be assessed for the imaginative use of the computer programs led to an abstract theme of 'a tower' with no specified constraints. The Participants were given some examples of famous 'towers' (**fig.1,2,3**) and were instructed to bear in mind that each design was to be completed within a time period of 30 minutes. In each case the subjects were carefully observed.



**Fig. 1 Tower of Pisa**



**Fig. 2 Eiffel Tower**



**Fig. 3 The Gherkin**

For the first phase of the experiment the designers were instructed to use a CAD platform of their choice to produce the first design. The use of CAD media was set out to be at the first stage of the experiment because, as all the participants were most comfortable with this tool, it was judged that it would be a smoother and 'easier' way to begin the design process.

The second phase of the experiment demanded programming skills from the designers. They were required to produce the program that would produce the design of a 'tower'. It was clarified, though, that the limited time

and programming skills of the participants justified the use and development of older 'code snippets' and acknowledged it as part of the programming process.

For the third phase the designers were provided "Faulty Towers" plug-in to AutoCAD developed by Christian Derix. This 'self organizing' tool is at an experimental stage and allows architects to quickly produce a *Faulty Tower*, do a sunlight analysis on it and run a genetic algorithm to produce improved generations of it. The designers were given a sort brief by the conductor of the experiment (the author) and the opportunity and extra time to familiarise themselves with the new tool. At this stage, mainly due to time constraints, only a small fraction of the versatility and power of *Faulty Towers* was made to use of the designers. Instead, it was 'misused', for the needs of the experiment, as a design exploration tool to produce forms that can be further developed. The reason that this tool was used last in the design process of this experiment was because it was judged by the author that this tool was more likely to affect the participants' concept and lead them to imitating the generated forms in the other two methods. Therefore, it was not introduced to them until they had finished with the two previous designs.

### 1.3 The concept of the experiment

The idea behind this exercise is that the spatial representations of such an abstract subject are, in themselves, meaningful representations, containing manifold interpretations, particularly associated with the use of advanced media.

The goal of this experiment is to gain insight into computer- human design interaction and explore the cognitive processes involved in acts of creativity. The generated objects are not meant to be inhabitable or functional but are explorations of 'reflection-in-action' between a designer and an interactive computer code. The use of different media intends to research whether the basic principles that define the design process change from one technique to the other and explores the alterations that the means cause to the dialogue between designer and design.

This research takes it as granted that the transition from the 'pencil and paper' to a variety of graphic systems has already begun and is now at a second stage of transition towards genetic techniques, such as the use of self- organising programs or the development of generative programs within programming environments.



## 2. Case Studies:

---

In the following unit, there will be an effort to categorize the final designs produced by the participants. Their evaluation will be based on the degree of influence by the computer program as well as the cognitive processes that the designers followed during the design process, as this was observed by the author during the experiment. The design process followed for each design tool will be commented and assessed separately and an overall comparative assessment, at the end, will point out some aspects that affected the set of designs.

At this stage, the results follow the grounded theory style analysis "*based on the premise that theory at various levels of generality is indispensable for deeper knowledge*". Additionally, the experimental data is equally important with the experiential, as likely to bias the research. Obviously, because of the nature of the data (experimental and experiential) to be analysed, the researcher's will not be the only possible interpretation. [Strauss, 1987, pp.6-11]

## 2.1 Designing with CAD

### 2.1.1 The designs

The designs realised through a CAD platform will be divided into two groups. Since during the briefing of the experiment it was not specified whether the CAD designs should be in 2-D or 3-D, there was a multiplicity of design methods. The tools that were used at this stage were AutoCAD (2-D or 3-D), 3D Studio MAX, 3D Studio VIZ, Google SketchUp (3D) or the combination of two of the above, usually a 2-D with a 3-D tool. However, the designs will not be divided in a 2D and 3D category accordingly. The first group will consist of the seemingly incomplete designs while the second the fully developed.

The designs that belong to the first group (**fig.4**) have a tendency to structural detail and descriptive characteristics that suggest a building under the primary stages of construction. It is remarkable, however, that in most cases the design seems to be, or actually is, part of an incomplete project that does not relate to a specific concept nor hints an undermined idea of further development. In addition to that, the designs in a way resemble and in no case can reveal an outstanding architectural characteristic.

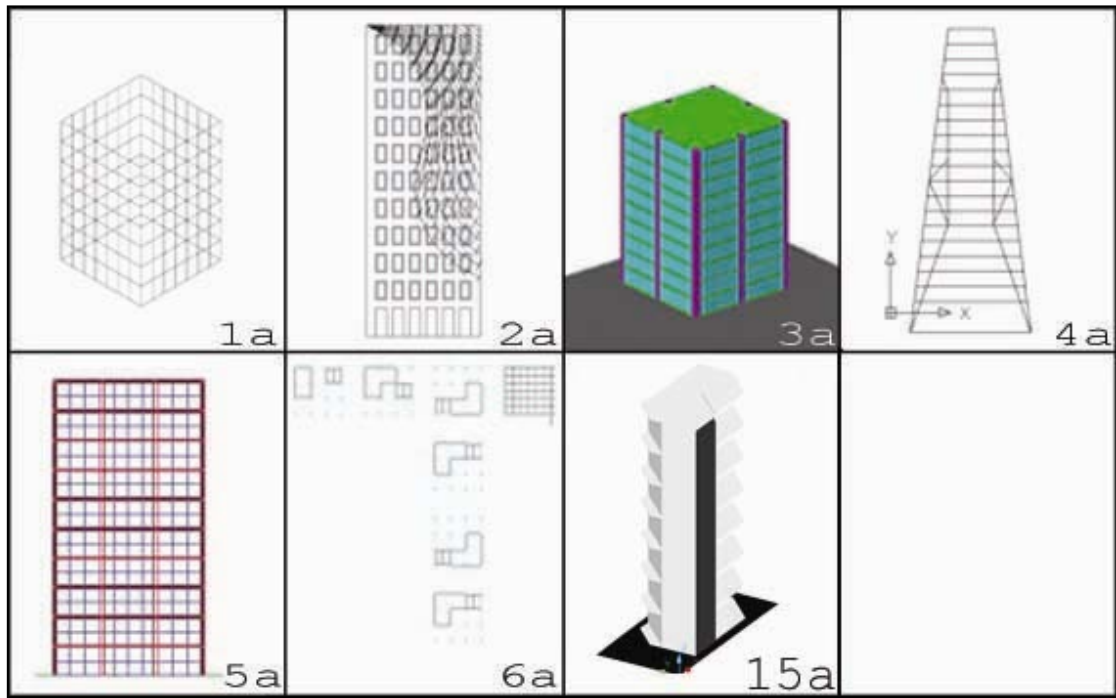


Fig. 4 CAD designs, group 1.

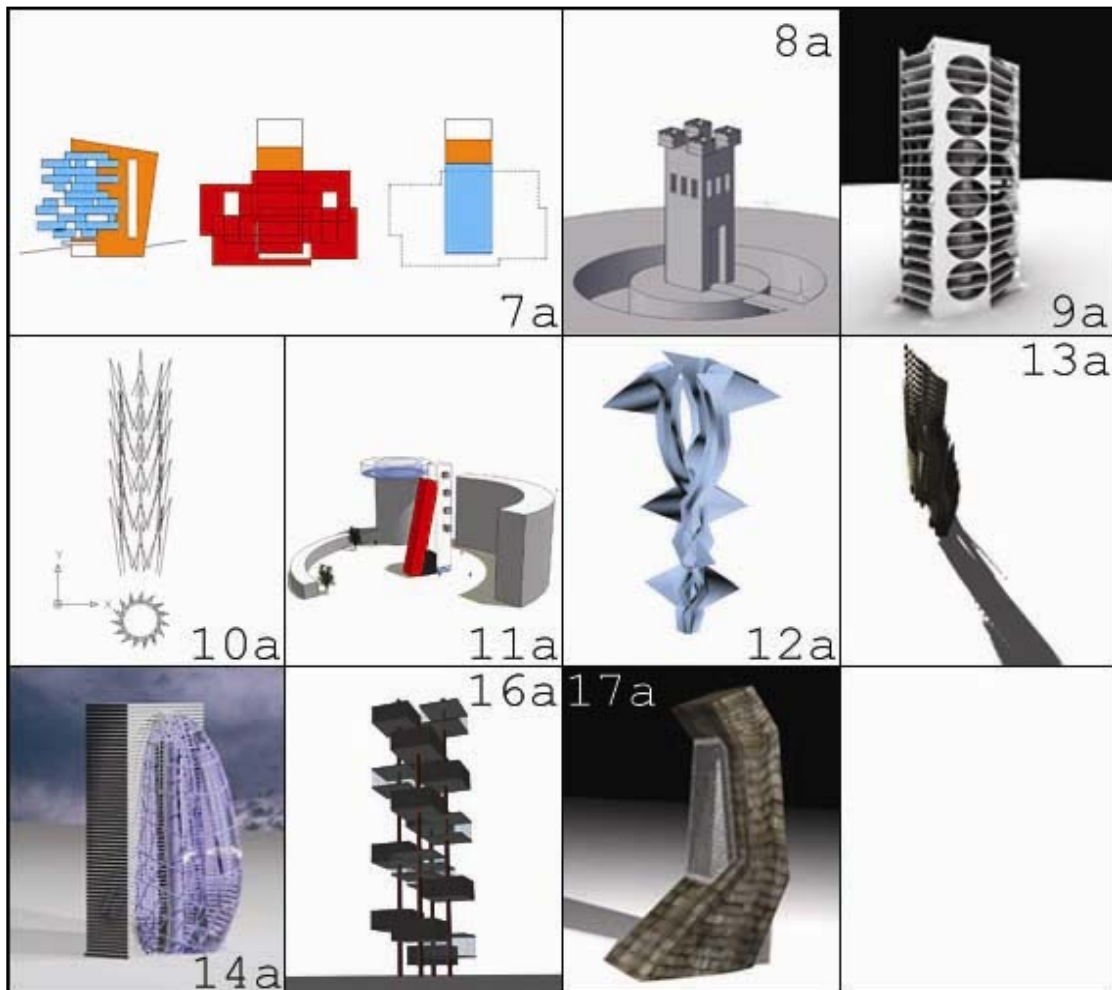


Fig. 5 CAD designs, group 2.

The second category (**fig.5**) represents the designs that have an underlying concept dominating them. The amount of detail or representational extension of each one depends on the skills of the designer over the medium as well as the speed in which they finalized their idea, in order to use the rest of the time to improve its representational features.

### 2.1.2 The designers

It was observed that the designers of the first category had a serious problem producing a design out of such an abstract theme and with no specific requirements or constraints. Most of these designers (**1a, 2a, 4a, 5a**) were totally abstracted by the lack of design program and focused on adding details to the design, handling it as an entity of its own, instead of the design representation of a 'tower'. The designers that expressed the greatest objections to the absence of a design program (**6a, 15a**) set the parameters on their own, used a paper and pencil sketch to finalise their idea and afterwards began to represent it in great detail through their CAD drawing. In these cases, the designers did not manage to complete their designs. At the end, they got very frustrated with the time limitation and returned to their sketch trying to explain to the observer what they meant to design. It is important to note that the designers of this category were not satisfied with the final outcome and found the design process of it rather dull (**fig.11**).

On the other hand, the designers that belong to the second category either had a very specific idea set in their mind while designing (**8a, 9a, 11a**), and therefore developed it quickly and used the rest of their time on perfecting the design, or used the computational tool as a design exploration platform. In all the cases, the designs were fully developed within the time limits and no complaints were expressed on this matter. The designers of this category all enjoyed the design process even though their opinion over the final outcome differs based on the aesthetic result. However, their opinion refers to the 'tower' and not to the quality of the design of the 'tower' [based on observations by the author and comments of the participants].

### **2.1.3 The program**

The use of a CAD platform for the realisation of the first stage of the experiment renders it very difficult to determine the level of effect that the program had over the final design. The fact that some designs resemble is not enough proof that the program affected the final outcome, since during the observations it was established that the designers used different methodologies for realising them. In addition to that, while all of the designers claimed to use the chosen tool very often and found its interface clearly structured and practical (**fig.11**), the majority of them used very simple instructions to perform the basic design and only higher level skills to improve its final representation.

It is necessary to point out that imaginative changes do not happen at random with the CAD. After all, the designers expressed that the outcome of their choices is predictable with CAD (**fig.11**). As it was used both as a representational tool and as a design exploration tool in multiple forms (2D, 3D or the two combined) to produce diverse results, it would be safe to conclude that the designers had a considerable control and choice over the design process and the final outcome they pursued.

## 2.2 Designing with Programming

### 2.2.1 The designs

During the second stage of the experiment, the majority of the participants used the Processing programming language and environment and, in particular cases, MATLAB or AutoCAD VBA Source Code. Again, it was not specified to the designers which language to use, since this was not important for the nature of the experiment. The designs generated through programming will be divided into two groups. This division involves the nature of the programs developed by the designers. The first group will consist of the representational programs and the second of the generative ones.

The designs that belong to the first group (**fig.6**) are developed through instructions that create the sub-shapes that comprise the 'towers'. These programs produce an identical result every time they are run. In some cases, the programs have some level of interactivity mainly due to a mouse or camera movement (**4b, 11b, 15b**) or a spinning of the design (**2b**).

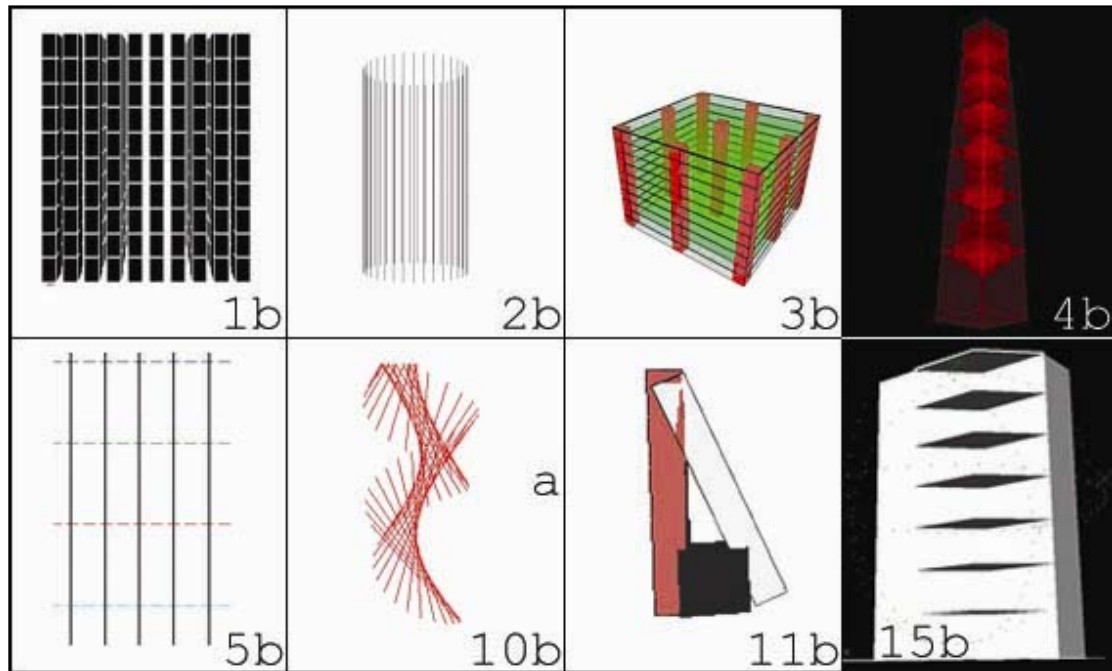


Fig. 6 Programming designs, group 1.

The designs that belong to the second group (**fig.7**) are generative, which means that the designers created a descriptive code based on a system of parameters, that they have set, which adapts itself each time it is run to produce one out of multiple solutions, expressed with a model that applies to the design problem. These designs can be from very simple randomization solutions to complex simulation systems that have been applied to the 'tower' problem. It would be worth mentioning in more detail some of them, but, the code itself was not the requirement. It is notable, that despite the fact that the designers were encouraged to combine multiple media during the design process- mainly supported by programming- only two chose to do so (**13b, 17b**), and only one (**13b**) did it to support the design process and not the presentation of the final outcome.

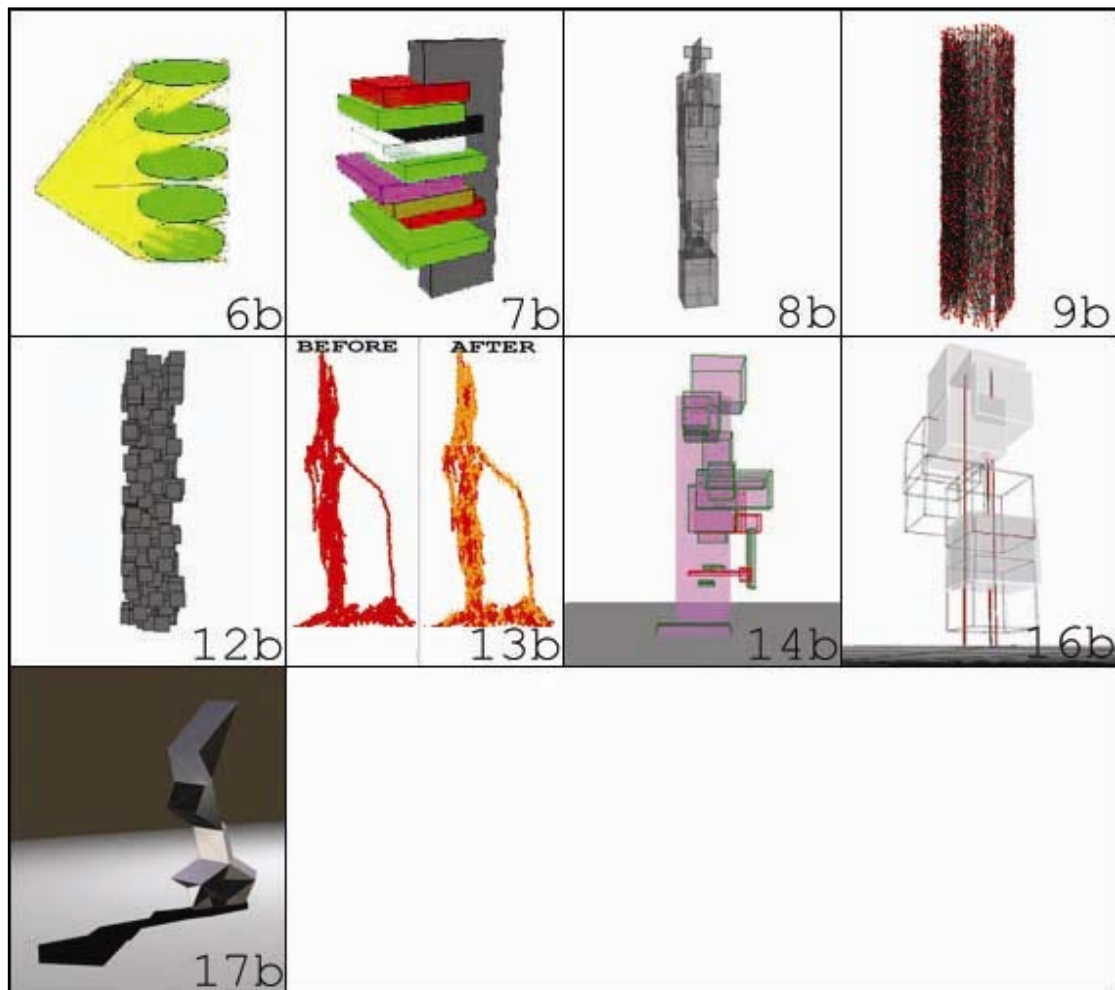


Fig. 7 Programming designs, group 2.

### 2.2.2 The designers

It is important to clarify that the final design is not judged for the scripting skills of the creator- even if this is an important factor. It has been observed that participants with impressive previous pieces of work have acknowledged, at the end of the experiment, to have produced a fairly simple and un-imaginative program for this exercise. It should be fair to mention, however, that some designers began scripting in an empty programming page, while others, used previously developed



programs that they adapted for the needs of this exercise. After all, they all admitted often using programs or actually some pieces of code acquired, as they stated through the questionnaires, mainly from the INTERNET (17/17) or colleagues, teachers and tutors (4/17), even books/manuals (2/17). It was made clear, therefore, to all of the participants from the beginning of the experiment that this was an acceptable procedure since the programming stage of the experiment frightened most of them, especially those who were not so skilled or confident programmers. The aim of the experiment, at this stage, was not to judge the scripts produced, but, the design process, the creative choices taken during this process in accordance to the final representation of the 'towers'.

At this level of assessment, therefore, it has to be mentioned that the designs that belong to the first category do not exhibit imaginative resolution towards the theme. In 7 out of the 8 designs of this group, there was a mere effort to reproduce the CAD design. Additionally, the designers performed the design process in a procedural way to merely get the result, without enjoying it.

The designers that belong to the second group created more abstract results focusing their interest on the generative process- fascinated by it- instead of the design itself. After concluding their program, they concentrated on adapting the parameters to produce an outcome that would be compatible both to the project's requirements and their aesthetic values. At the end, they run the program many times to choose a design that they would submit as final. An objective observer would

comment that many of the designs of the second group (**8b**, **12b**, **14b**, **16b**) bear similar properties. However, it is impressive to see the variety of methodologies that produce these designs.

### 2.2.3 The program

The use of programming for the realisation of the second stage of the experiment proved to be very frustrating for the designers. This was mainly due to their lack of confidence. However, the results did not seem to be affected by this fact since designers, highly stressed at the beginning of the process, produced quite satisfactory results. The diversity of both the methodologies used and the designs produced, proves once more that whatever difficulties the designers might have expressed, at the end, their choices led to some outstanding results, especially considering that most of them only learned programming in the previous year. However, in spite of the fact that the majority of them found the design process creative, they admitted having some trouble predicting the outcome of their scripting, mainly because of the generative nature of some programs or their limited skill (**fig.11**). Anna Laskari, an architect who participated in the research admitted that "unpredictability lies within limits, set by the structure of the program. For example, the script I used for this research gives rather unpredictable results because it is based on randomised functions".

## **2.3 Designing with a Self-Organizing system**

### **2.3.1 The designs**

During the third stage of the experiment the *Faulty Towers* plug-in to AutoCAD developed by Christian Derix, was used. This time, because of the difficulty to find a self-organizing system and for the need of homogeneity of the results, the participants were all obliged to use this specific tool. This tool when it is run, under a set of parameters specified by the user through a defined interface window, will build a 'faulty tower' which will be the seed for an initial generation production. This initial generation will be improved through an algorithmic process where the user is responsible for selecting a match-parent for the fittest tower of each generation.

At this stage it needs to be clarified that the experiment does not aim to research the degree of interactivity and interference that the designer has with this specific generative tool. It researches whether the designer can claim any level of authorship through the whole design process- aided by this tool- over the submitted final design. It was judged, therefore, that because of the homogeneity of the 'faulty towers' produced, clearly referencing to the aesthetical choices of the designer of this tool, the design cannot exhibit at this stage any authorial control of the user. Consequently, the results will be divided into two groups according to the level of interference that the designer exhibited towards their chosen 'faulty tower'.

The first group (**fig.8**) includes the designs that have none or minimum level of interference. In this case, the final designs resemble to each other within a level of variation. The efforts that were made to transform these results were confined to the deletion (**2c, 8c, 10c**) or transformation (**12c**) of component elements and the simple addition of materials to an unchanged 'faulty tower' (**13c**).

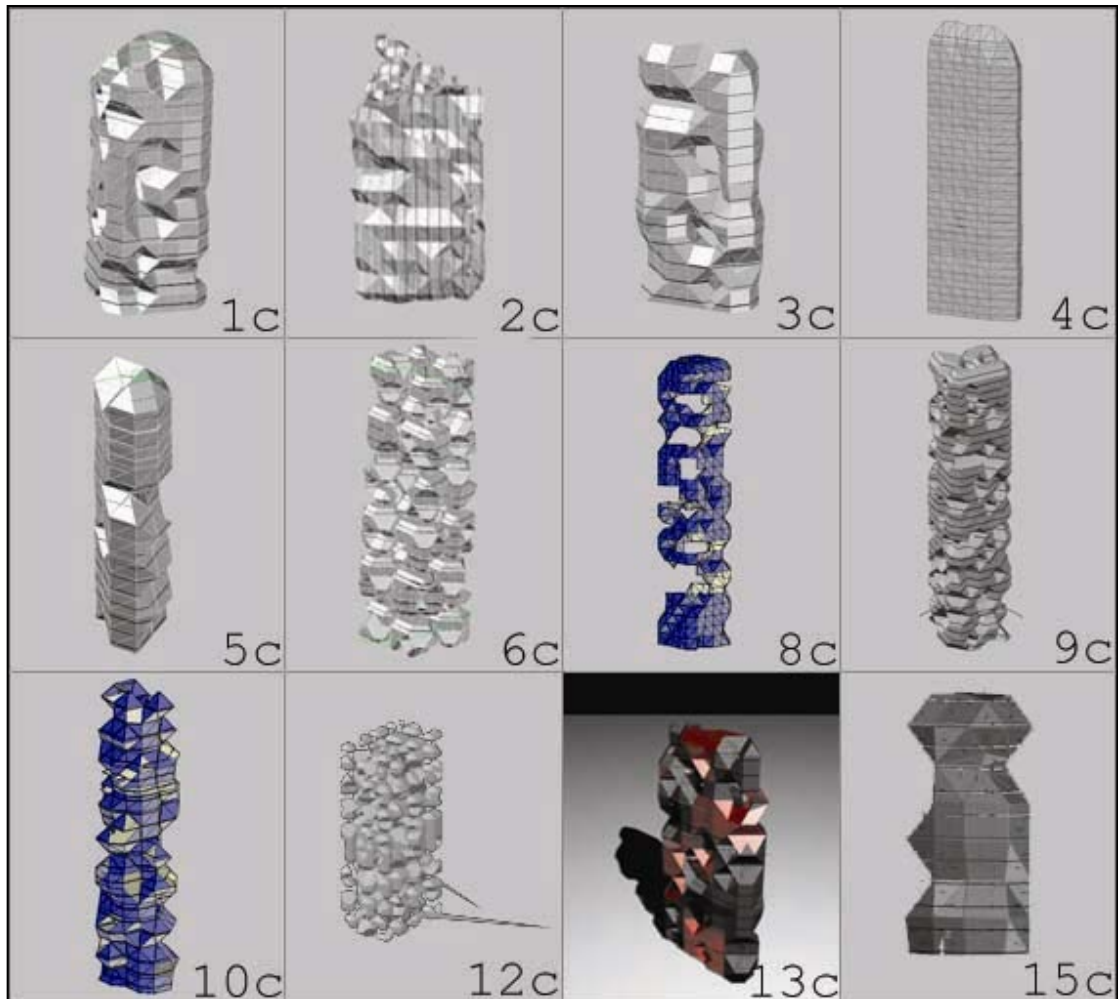


Fig. 8 'Faulty Towers', group 1.

The second group (**fig.9**) includes designs that have suffered more radical, compared to the others, or rather interesting transformations. These 'towers' have component elements that remind an observer that the designer has used the specific tool but there is also an underlying concept for each one that makes it differentiate from the others. At the same time, in some cases (**7c**, **14c**, and **16c**) it is judged that the designs could be easily reproduced without the use of this tool. It is interesting that only one model (**11c**) combined more than one 'faulty towers' to produce the final 'tower'.

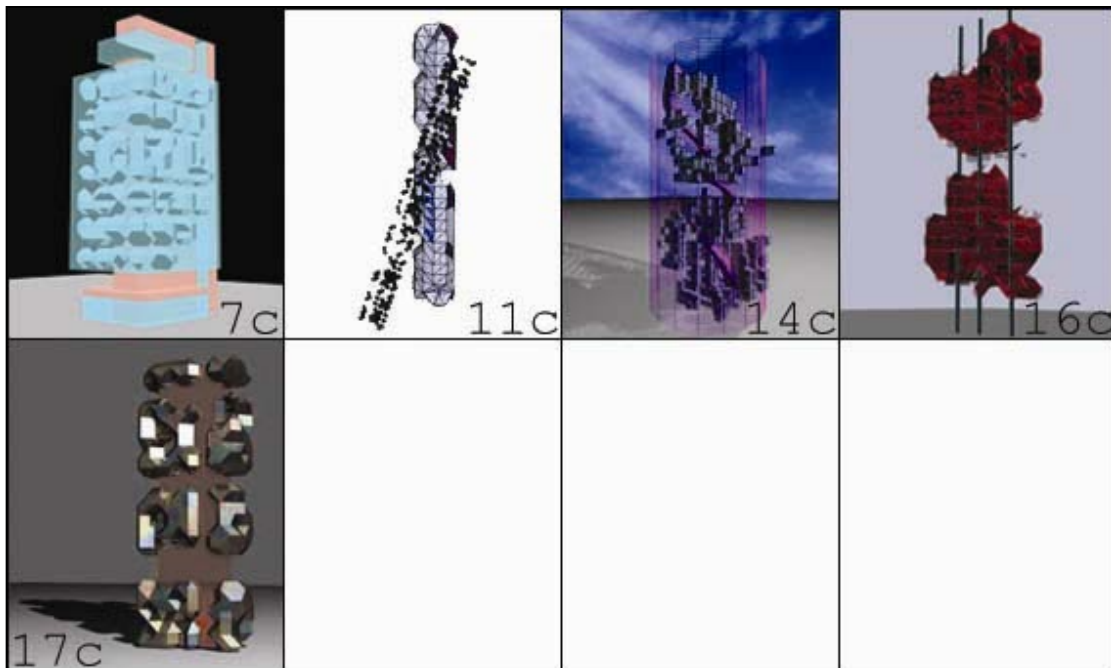


Fig. 9 'Faulty Towers', group 2.

### 2.3.2 The designers

The fact that the designers were obliged to reproduce the process within a short time of learning about it was a factor that obviously affected the final outcomes. However, since most of the participants in this experiment come from an architectural design background, they are used to assimilating knowledge intuitively, and also to 'learning through doing'. After all, architecture teaching has always been practice-based learning and the design process has been, as characterised by Schön [1987], 'reflection-in-action'. The existence of some extremely interesting design solutions, like the ones of the second group, supports this theory. Additionally, it can be regarded safe to draw conclusions about their outcomes because all the participants have exactly the same amount of experience with the tool. In other words, any variations will not be due to more experience or greater skill.

It is important to be noted that, at the beginning of this stage of the experiment, it was pointed out to the designers that this tool was merely a plug-in to AutoCAD. Its process could be terminated at any stage that the designers wished to and the designs could be processed as a standard AutoCAD 3-D model. In accordance to that remark, one design (**14c**) did not use one of the 'faulty towers' produced by the tool. Instead, it developed the cubic configuration resulting out of a matrix relation that it produces at the early stages of its run, before building the actual 'faulty tower'. It is interesting to highlight that the designer that pursued that choice felt obliged to ask the permission of the observer to do so, even if it was requested of her to use

the tool in any way she judged would be most helpful for the realisation of her 'tower'.

### 2.3.3 The program

It proved to be extremely difficult, at this stage, to define after what level of interference the designs were detaching from the tool and its designer. The designers found it quite difficult to predict the outcome that it produced and, thus, their understanding of the way the parameters that they changed affected the final outcome was limited (**fig.11**). Adam Doulgerakis, an architect- designer of a 'faulty tower' for this experiment, pointed out that *"the design with such a program demands that the designer should be involved with the rules that generate the outcome per se. Thus, it is rather difficult to produce meaningful outcome unless the user is the software designer himself, or very familiar with the function and the structure of the program"*.

In comparison to the previous stages of the experiment, this tool is undoubtedly affecting the produced designs in a deeper level. It cannot pass unnoticed, after all, that the designs that exhibit some level of imaginative initiative on behalf of the users, like the ones of the second group, are a minority. The architect-participant Rafail Bakolas commented: *"[Self organizing systems] limit, in a way, the final result in terms of innovation, if used without wisdom and if the designer- possessed by the usefulness of the tool- fails to evaluate the output correctly"*.

## 2.4 The 'digital' design process: General Observations and Comments

*"It is true that, especially in the case of architectural design, constraints are far from being the antithesis to creativity. On the contrary constraints on thinking are what make it possible."* [Boden, 1990, p.82]

In this experiment, it was judged preferable that the constraints would not be set at a requirement level rather than in the context of the computer program. This decision seems to have affected some designers greatly (9 out of 17). Chris Leung, a research engineer who participated in the research, commented: *"Defining constraints and objectives [...] is necessary and highly desirable"*. This fact was intensely observed at the first stage of the experiment and much less at the other ones. This, probably occurred because the designers felt more familiar with, and mainly, less constrained by the CAD programs. Subsequently, they felt a greater need for a more specified theme.

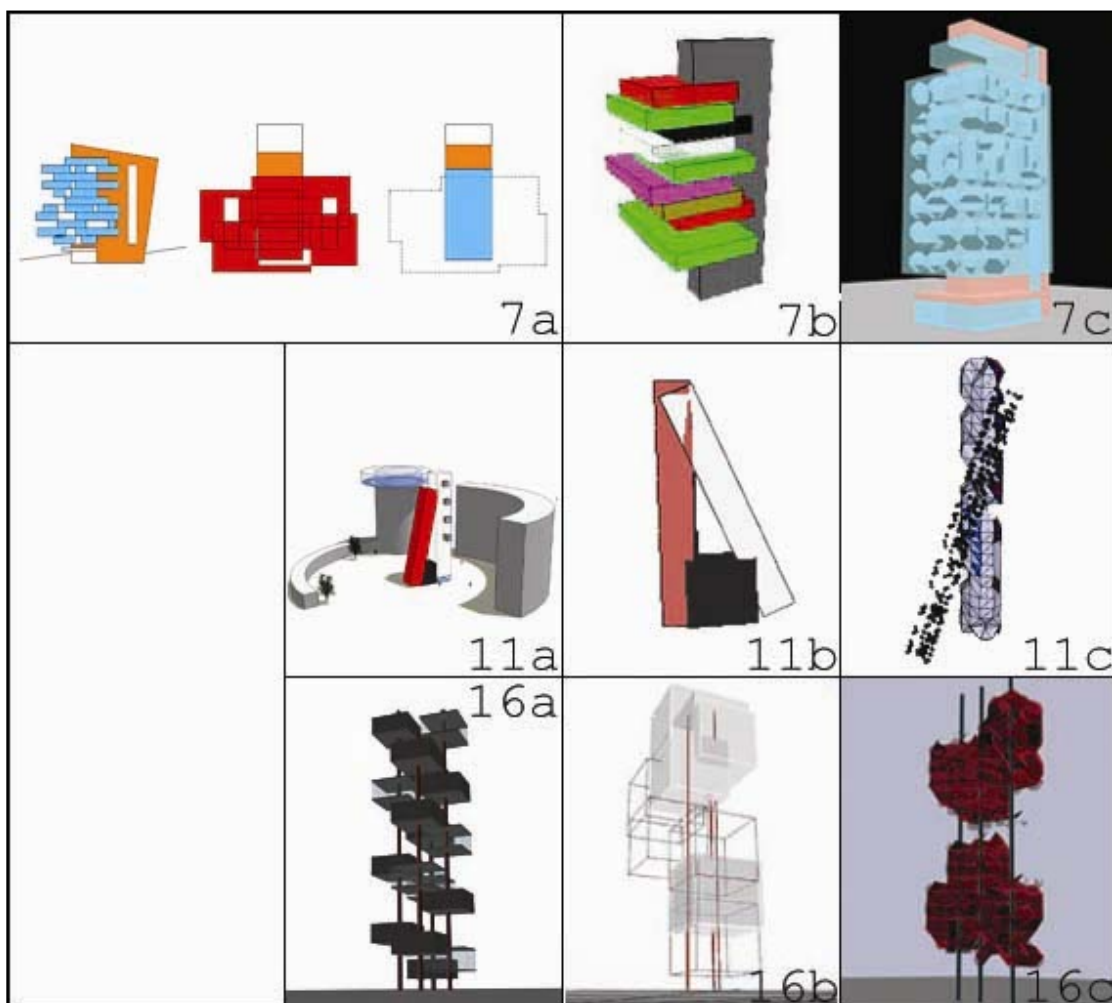
In contrast, when those same designers passed onto the other methods, they did not seem much preoccupied with the theme as much as they did with its realisation through the given platform, thus, forgetting the lack of detailed requirements. It is interesting to know that this particular group of designers when reaching the third stage of the experiment, all relieved, presented a 'faulty tower' unchanged as a final design. However, two of the complaining designers (14, 16) decided to use the platforms provided as design exploration tools and their results are extremely interesting.



A common question, asked by the designers at the beginning of the experiment, was whether it was required of them to represent the same concept- idea through the three different methods or it should be a different concept for each medium. This was a question that was deliberately not answered in a definite way; instead, it was left upon the designers to decide which route they would follow. The reason for that abstraction was that the designers would be assessed for the choice they would make on that matter. It was indeed interesting to observe that, at the first phase of the experiment, more than half of the participants were determined to set an idea and follow it throughout the experiment (9 out of 17). This methodology seemed either easier to them or more logical at that time, since the requirement was 'a tower by three computational methods'. However, when it came to transferring the, often complex and detailed, designs of the first stage into the self organizing system or through programming, in most cases the participants changed their mind.

In the case of programming, the participants followed two methods of designing. Most of them (1, 3, 4, 5, 11, 15) used their program similarly to the CAD platform. They soon realized, though, that compared to the CAD, inserting a great level of detail to a programming design is both time consuming and dull in a repetitive way. For this reason they simplified or modified their designs accordingly. The rest of the designers (2, 10, 16), on the other hand, did not adapt just their idea but their designing logic as well. They pursued a genetic logic for the designing of their 'tower' and produced comparatively more interesting results.

In the case of *Faulty Towers*, and after the trial runs during the familiarizing stage, most designers concluded that it would be impossible to pursue the 'one idea' logic. Only three designers (10, 11, 16) out of the nine achieved the goal that they set at the beginning of the experiment (**fig. 10**)



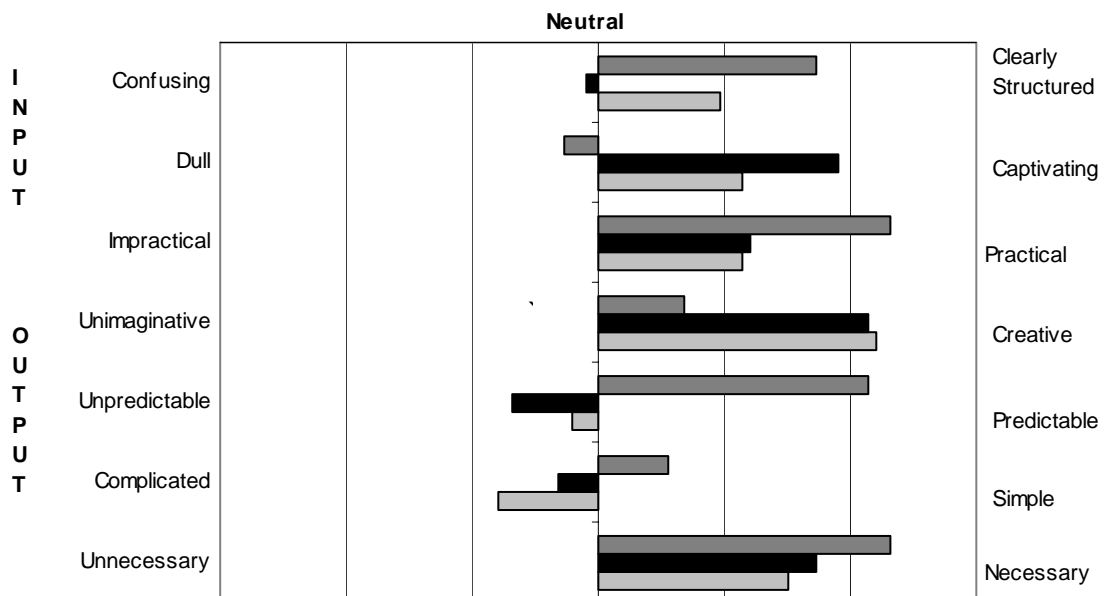
**Fig. 10** Projects with one concept throughout the experiment.

### 2.5 Authorship

After the experiment, the participants were inquired through questionnaires about the way they perceive the authorial role of the architect, in relation to the digital design media. The purpose of this inquiry was to depict the opinion of young architects currently in avocation with the latest media- a fact that yet cannot be expected of all architects. Additionally, having experienced the design process through three different computational techniques, it is interesting to compare their opinions on these interactive processes.

#### Designers' View of the Methods

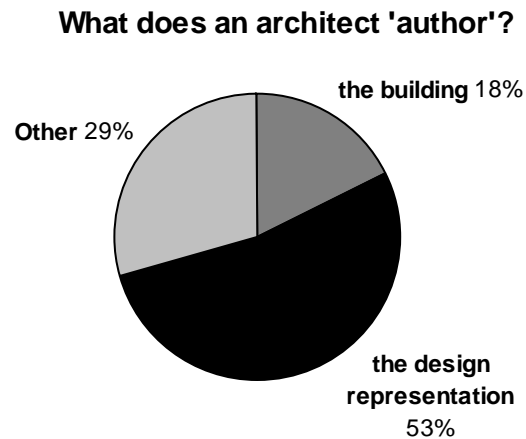
□ Programming    ■ Self Organizing    ▒ CAD



**Fig. 11** Representational graph showing the designers' views on the computational methods used at the experiment.

In general, the designers expressed that the CAD platform is considerably more practical than the other two computational means, but, far less creative and rather dull, as well. In the case of programming, the fact that its interface is quite clearly structured does not diminish the complexity and unpredictability of its result, based mainly on the generative properties that the program might have. The design process through a self organizing system was judged to be captivating and with particularly creative results. The fact that the causal nexus was not clear and the results seemed difficult to foresee affected the design process and intrigued most of the designers to discover its underlying processing structure (**fig.11**). Sonam Gupta, one of the 'designers', commented on the matter: "*Self organizing programs open a plethora of options 'IF' we understand the parameters and their rules of working- which is very difficult.*"

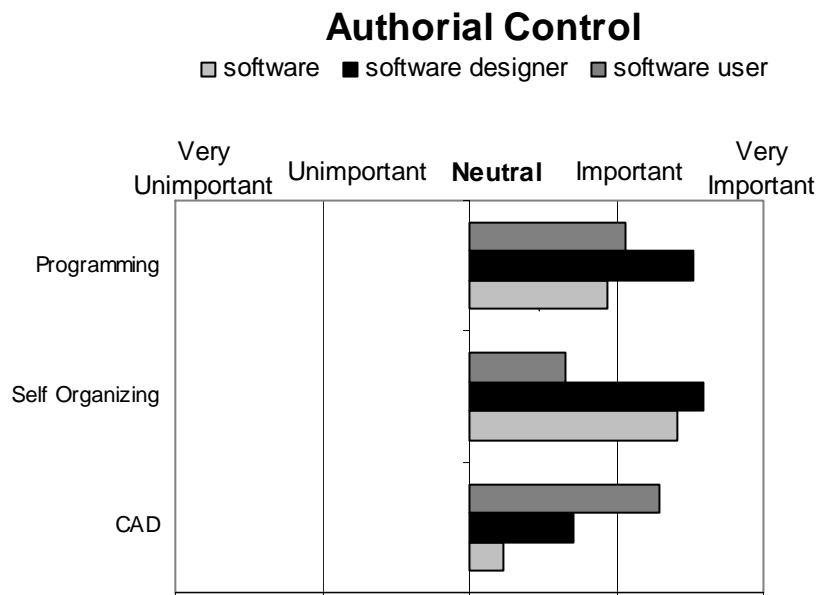
Moreover, the participants were asked whether they are familiar with the notion of authorship and all but one answered positively. Nonetheless, when those same people were asked what an architect 'authors' they were deeply puzzled. Half of them answered that it is the design representation of the building and not the building itself. But a considerable amount, replied that it is both, because what the architect actually 'authors' is the idea, the design concept which is expressed through the building and its design representation (**fig.12**).



**Fig. 12** Graphic representation of the participants' answers about what an architect 'author'.

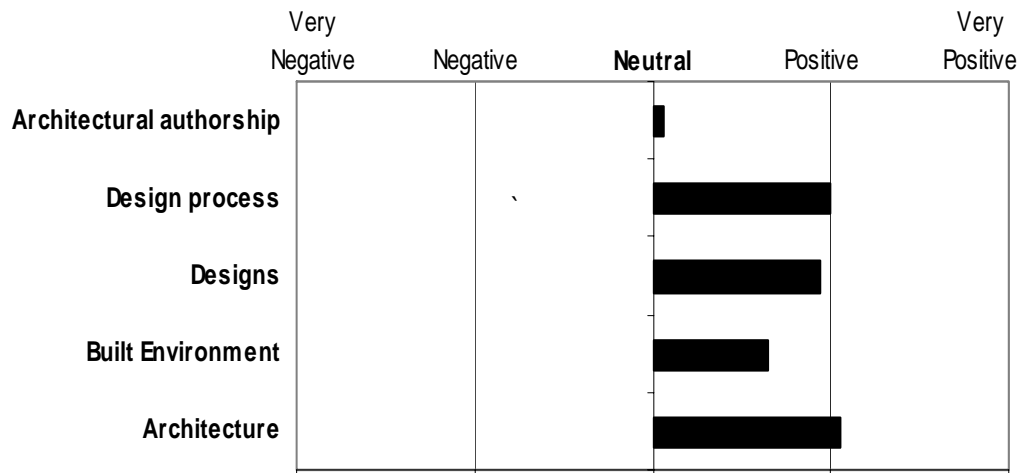
The authorial control, which the designers considered having in comparison to the software designer and the software itself, was judged to be considerably important with the CAD but gradually diminished with the use of a programming environment and then the self organizing system (**fig.13**). Tassos Kanellos commented on the use of the last two techniques: *"The authorship of any design will still belong to the designer- architect but only part of it, since the final design outcome will depend on the efficiency of the program. The part belonging to the architect will have more to do with the creative process of establishing concepts and the framework, where the program can be of aid. As it is difficult for a single program to be able to aid any design process, the architect's ability to understand and adapt programming code becomes rather important."* And Anna Laskari, another 'designer', added: *"The degree of authorship from the part of the user depends on his understanding of the process that leads from the input to the output"*

On the contrary, the role of the designer of the self organizing system and in the case of programming the designer of the 'code snippet' that the user is adapting is interestingly improved compared to the CAD. The authorial control of the program is in all cases less important than that of its designer but not always than that of the user's. (**fig.13**)



**Fig. 13** Graphic representation of the participants' views on the authorial control that the software, its designer and user have over the final outcome.

To sum up, the designers could not really agree on whether the possibility of 'self organizing' systems is positive, negative or neutral. Opinions on that matter are controversial and the issue attracts a lot of debate. However, as far as the design process, the designs, the framework of architecture and its expression through the built environment are concerned, the general impression is positive. (**fig.14**)



**Fig. 14 Graphic representation of the participants' view on the influence of the use of 'self organizing' systems in the architectural process.**

Throughout the experiment the results were very interesting and in some cases surprising. However, it is necessary to relate these results to the theoretic review that was laid in the first part of this thesis. In that way, it will be shown how the four concerns raised at the beginning of the experiment (the cognitive processes followed by the participants during their interaction with the digital media, the control that the participants had over the various computational methods and the final designs, the control of the tools used over the participants and the final designs and the flexibility of the participants to mixing different digital media) can be compared with the relative theories, for a general consumption to be drawn. Out of this assumption, this thesis is aiming to reveal the authorship relations within the 'digital' design process.

## Discussion

---

In this chapter the findings of the experiment will be compared and discussed in accordance to relative theoretic academic work.

### The Architect-user

*"Designers do many things that computers don't. Some of these are bad habits that the stringencies of computation will correct. But others are basic to design, and cannot be ignored if computation is to serve creation and invention."*

[McCullough, Mitchell, Purcell, 1990, p.17]

G.Stiny through these lines taken from his treatise on *"What designers do that Computers should"*, states that it is the correlation between computers and designers that is to promote the design process and not the predominance of one over the other. *"Being digital is different. We are not waiting on any invention. It is here. It is now. It is almost genetic in its nature, in that each generation will become more digital than the preceding one."* [Negroponte, 1996, p.231] The insertion of digital media has long been established within the architectural methodology; it is merely a matter of time to find the appropriate equilibrium that will relieve the architects from the anxiety of using computers and will transfer their focus from the medium to the process.



During the experiment conducted by the author, it was established that many of the designers felt frustrated when asked to complete a simple task with the use of a computational technique with which they were not particularly familiar. *"In general, designers seem to find this experience of using computers a frustrating one"*. [Lawson, 2006, p.282] However, the fact that, on the one hand, there were participants that felt stressed with the use of media they consider their indispensable tools, like the CAD, and on the other hand, participants that enjoyed the process of designing with a tool they had just been introduced to, only proves that the nature of the architectural process, characterised 'reflection-in-action' by Schon [1987], is capable of expeditiously adapting the means that are offered in order to serve creativity.

W.J.Mitcell in "The Electronic Design Studio" raises the topic of the aspect- blindness of the CAD systems naming in that way their weakness, in contrast to the human brain, to recognise emergent shapes within a system of multiple shapes. [McCullough, Mitchell, Purcell, 1990, p.5] However, during the third stage of the experiment, some of the participants singled out of the emergent shapes of the system component elements to use within their own creative ideas, while others, failed to see shapes other than the produced ones. Therefore, architecture is not afflicted with the aspect- blindness of the system but the aspect- blindness of the architect who selects and interacts with the system. *"Even if systems were coded for elegance at every step of the way, the result would be programs, not experiences, and interfaces, not interactions"*. [McCullough, 2004, p.12]

Hence, the question, as so elegantly Lawson put it, "is not so much about what happens inside the computer but how we converse with it". [Lawson, 1990, p.128] Often, though, the complexity of the software exceeds the understanding capacity of the user (like with the self-organising program at the third stage of the experiment), thus cancelling the dialectic process between the two, leading to the domination of the first and, in consequence, the exclusion of the last. For that reason, it depends on the responsiveness of the architectural educational system and, above all, on the awareness of the designers to retain control of the process and not surrender full authority to the system.

Margaret Boden in *The Creative Mind: Myths and Mechanisms* recognises that, on the one hand, an autonomous system "is governed by a set of principles which uses a provided set of input to determine a function"; on the other hand, "the system lacks properties such as choice, discernment and judgment". [Boden, 1990, p.22] It has been established, over time, that various computational methods have achieved to be of aid to the architectural process, otherwise, most of the designers would not consider them that necessary (**fig.11**) and be eager to integrate them in their course of work. It is also undisputable, however, that technology has not yet reached the point to achieve 'artificial intelligence' the way that human intelligence is perceived and the "Architecture Machine" that Negroponte describes [1970] is yet to be developed. Thus the 'choice, discernment and judgement' are trusted with the authorial figure of the architect.

*"Creativity requires more than the mere automatic mixing of ideas. Even an unguided 'bisociation' of matrices is not enough"*

[Boden, 1990, p.23]

This thesis after analysing both the designs and the design process, as it was realized with the use of multiple digital media, is forced to conclude that the diversity of the final results cannot in any of the stages claim responsibility of the systems beyond doubt for either the imaginative or unimaginative results. Contrary, it was observed that the initiative and stance of the participants were the determinants of the formed outcomes. Consequently, the system itself has minimum authorial control over the design and merely threatens the authorial control of the un-initiative designers.

### The Architect- programmer

*"The programs are like surrealist paintings, which have both aesthetic qualities and technical excellence. Their work is discussed both in terms of style and content, meaning and performance. The behaviour of their computer programs has a new kind of aesthetic."*

[Negroponte, 1996, p.221]

The new hybrid of architects- programmers that is emerging has given architecture new dimensions. The designers don't wait for the programs to be developed so as to fulfil their needs; instead they built their own tools and they design their own programs. The architect has progressed from building the physical model in order to observe the "play of masses in light" [Mitchell, 1999, p.41] to 'breeding' the digital model.

In the first part of this thesis, nevertheless, a concern was expressed on whether the architects would be satisfied with acquiring the role of the 'breeder of forms'. In the previous unit, the effort was concentrated on proving that the architect as a user of generative techniques has responsibility per se to establish his authorial control. What is, though, the authorial control that the designer of the self-organizing, generative software has over the design produced by the software?

*"Every piece of software reflects an uncountable number of philosophical commitments and perspectives without which it could never be created. Software depends inevitably on our ideas about representation and reality"*

[Dourish, 2001, preface]

In the digital world that we live in, the 'code' is the law, so understanding and controlling the code means power. Through programming the designer acquires additional skills to perfect the design process; however, the essential first step of the concept is what forms the foundation of architecture. During the experiment performed for this thesis, the designers were asked what an architect really 'authors' and a significant amount of them answered the design concept, the idea (**fig.12**). The Albertian figure of the architect who, in divine inspiration, conceives the idea before setting to design is, therefore, hard to circumvent. In the programs built from the participants at the second stage of the experiment, among the generative ones- even the ones that had seemingly resembling characteristics- an observer could identify the designers' sought after qualities in designs.

Further more, at the third stage, most of the users failed to build a tower that would reflect them, because the 'faulty towers' were already characterized by the aesthetic choices of the creator of the tool. The ones that managed to maintain authorial control over their designs were the designers that had a concept of their own and used the component elements of the 'faulty towers' to realize it, or, the designers that used it merely as an inspirational tool in order to develop a new concept. The one designer that decided not to complete the process of the system, felt obliged to apologise to the obscure software designer. It is interesting that when the designers were asked about the authorial control of the designer and the software, in the case of the self

organizing system almost all of the participants identified them as one unit. (**fig.13**)

It is, therefore, the conceptual model of generative tools which is important, not the hardware or software. The architect-programmer is more than the 'breeder of forms'. He is engaged in a new form of design process that is expressed through the programming language and is represented through the solutions that the program provides. After all, architecture is not really about designs; the designs are merely the representations of the architect's creative ideas.

### **Further thoughts**

It is unquestionable that the number of participants in the experiment was not large. The truth is that the architects-programmers are for the moment a minority of the architectural community. In a further work, a larger sample of designs produced by various computational methods and a larger scale inquiry on the subjects under consideration in this thesis would produce more statistically accurate results.

The ideas presented in this thesis are merely the tip of the iceberg of an ever evolving architectural field. Nevertheless, an in depth research on the way the state-of-the-art techniques are integrated in the design process and influence issues of authorial control, is needed for a smoother transition to the computer-aided architectural practice.

## Conclusion

---

Neisser [1967] devotes his book *Cognitive Psychology* to establishing that humans are different from machines from the very beginning of the perceiving and thinking process. Information processing machines have generated a new perspective on human thought but cannot actually replace human thought.

This thesis was set to re-establish the authorial control of the architect as he integrates the digital media in the design process. Although it has been frequently analysed in the architectural community how the emergent generative techniques exalt architects and architecture into new paths of creation and creativity, it is not yet fully established whether the produced forms are the creations of the computer, the architect-user or the architect-programmer.

In the first part, the design process was explored through the cognitive processes of the human nature, the emergence of evolutionary techniques in architecture was revealed and, finally, the authorial role of the architect in the course of time was presented. The theoretical review of the first part of this paper set the issue of authorship and architecture under consideration. An experiment, thus, was conducted by the author aiming to identify in practice the authorial control of the designer through different computational design techniques. In the second part, the results of this experiment were presented and analysed in detail and an effort was made to draw some conclusions about the designing behaviour of the participants and their

manipulation by the programs. After the findings of the experiment were related to the theoretic views on the topics under consideration at both conceptual and practical levels, it was concluded that the architect—either as a user of various computational techniques or as the designer-programmer of genetic tools—is maintaining the authorial control as long as his creative ideas order the machine and it is not the intelligible principles of hardware and software that advice his ideas.

Ultimately, the architect similarly to the practitioner of Schön has at his disposal “a repertoire of media which enables him to choose the graphic system best suited to the exploration of particular phenomena. Sketches enable him to explore global geometries; cross-sectional drawings, to examine three-dimensional effects; drawing to scale; to experiment with the dimensions of designs; models, to examine relationships of building masses, comparative volumes, sun, and shade. He uses media selectively to address the issues to which he gives priority at each stage of the design process.”[Schön, 1991, p.158] Eventually, the ‘code’ is one out of the many tools that the architect—author masters in order to produce architecture.



## References

---

1. **Alberti, LB, 1485**, *De re Aedificatoria* (Florence), "*the Art of Building in Ten Books*" (1988), trans. Rykwert J, Tavernor R, Leach L, Cambridge Mass., London.
2. **Anstey, T, 2005**, "The ambiguities of disegno", *The Journal of Architecture*, 10(3), pp. 295-306.
3. **Anstey, T, Grillner, K and Hughes, R (edit.) 2007**, *Architecture and authorship*, Black Dog Publishing Limited, London.
4. **Barthes, R, 1977**, "The Death of an Author" in *Image-Music-Text*, Stephen Heath trans. and ed., Fontana, London, pp.142-148.
5. **Boden, MA, 1990**, *The creative mind: myths & mechanisms*, Weidenfeld and Nicholson, London.
6. **Borges, JL, 1970**, 'The Library of Babel' in *Labyrinths: Selected Stories and Other Writings*, Penguin, London.
7. **Branko, K, 2003**, *Architecture in the digital age: design & manufacturing*, Taylor & Francis, London.
8. **Cross, N, 2001**, "Can a machine design?", *Design Issues*, 17(4), pp. 44-50.
9. **DeLanda, M, 2002**, "Deleuze and the use of the genetic algorithm in architecture", in N.Leach (ed.), *Designing for a digital world*, Wiley, New York.
10. **Dourish, P, 2001**, "Where the action is: the foundations of embodied interaction", MIT Press, Cambridge (Mass.), London.
11. **Frazer, J, 1995**, *An evolutionary architecture*, Architectural Association, London.

12. **Habraken, NJ & Gross, MD, 1988**, "Concept Design Games", *Design Studies*, 9(3), pp. 150-158.
13. **Hillier, B & Leaman, A, 1974**, "How is design possible?", *Journal of Architectural Research and Teaching*, 3(1), pp. 4-11.
14. **Lawson, B, 1990**, *How designers think: the design process demystified*, 2nd edition, Butterworth, London.
15. **McCullough, M, 2004**, *Digital ground: architecture, pervasive computing and environmental knowing*, MIT Press, Cambridge, Mass., London.
16. **McCullough, M, Mitchell, WJ and Purcell, P (edit.) 1990**, *Architectural knowledge and media in the computer era*, MIT Press, Cambridge, Mass., London.
17. **Mitchell, WJ, 1999**, *E-topia: "urban life, Jim- but not as we know it"*, MIT Press, Cambridge (Mass.), London.
18. **Negroponte, N, 1970**, *The architecture machine: toward a more human environment*, M.I.T. Press, Cambridge (Mass.), London.
19. **Negroponte, N, 1996**, *Being digital*, Vintage Books, New York.
20. **Neisser, U, 1967**, *Cognitive psychology*, Englewood Cliffs, London.
21. **Plato, 380BC**, *ION*
22. **Schöen, DA, 1983**, *The reflective practitioner: how professionals think in action*, Harper Collins, New York.
23. **Schöen, DA, 1987**, *Educating the reflective practitioner*, Jossey-Bass, San Francisco.
24. **Sternberg, RJ (edit.), 1988**, *The nature of creativity: contemporary psychological perspectives*, Cambridge University Press, Cambridge.

25. **Strauss, A 1987**, *Qualitative analysis for social scientists*, Cambridge, England: Cambridge University Press.
26. **Thompson, D'Arcy W, 1961**, *On growth and form* (abridged edition), Cambridge University Press, Cambridge.

## Appendix I: Participants' Comments

---

In this section the comments of some of the participants are quoted.

"Self organizing programs open a plethora of options 'IF' we understand their parameters and the rules of working- which is very difficult. They enable to combine and automate various spatial aspects of a building without the user needing to get into the actual calculations of each instance"

[Sonam Gupta, architect]

"Using advanced computational methods to automate certain design procedures, in order to quickly get designed results, sometimes enhances the design process enabling the architect to test several different (similar or not) outputs. Similarly to other approaches, on the other hand, they limit, in a way, the final result in terms of innovation if used without wisdom and if the designer- possessed by the usefulness of the tool- fails to evaluate the output correctly, a process that usually takes time to master."

[Rafail Bakolas, architect]

"Using that kind of self organizing systems in the design process, the architects' creativity is diminished. Their role is not as active as when using traditional design methods. On the other hand, the forms that derive from such processes are unexpected and even surprising."

[Antonia Mauroudi, architect]

"Using autonomous aspects in generative process resets the role of the architect as designer in the design process- positively. It reveals other possible variations of design that, otherwise, would not be revealed to the architect."

[Martin Kaftan, architect]

"Using Programming codes for design can perform designs based on engineering calculations while, procedures that embody more creative theories, such as self organising systems, produce designs based on architectural inspirations rather than calculations. Use of CAD packs just put down on the drawing sheet the predetermined inspiration of the designer. The combination of the three designing procedures could optimise the process starting from a self organising system to formulate the shape of the outcome, continuing with a source code program to verify that the required calculations could be achieved by the initial thought and a CAD package to produce the standardized drawings."

[Athanasios Kolios, Mechanical Engineer]

"The new computational tools provide a range of choices for the architect to select from during the different stages of the design process. In addition to that, architecture is about a network of relations between people, between various tool systems and between the people and the tools. The connections between the members of this network are recognisable and adaptable. "

[Shih-Ying-Chung Chou, architect]

"In general, 'self organising' systems exhibit creativity in terms that can provide the user/designer with a variety of otherwise unthinkable and unexplored solutions. Additionally, the design with such a program demands that the designer should be involved with the rules that generate the outcome per se. Thus, it is rather difficult to produce meaningful outcome unless the user is the software designer himself, or very familiar with the function and the structure of the program. On the other hand, whereas the standard CAD packages have usually less unpredictable outcomes, it is easier for the user to control the designing process as the elements of the software are the tools that compose the idea that emerges from the interaction of the designer with the software. If this interaction will be fruitful it depends on the software design and features that this particular package includes."

[Adam Doulgerakis, architect]

"In the specific self organizing program used in the research, the output seems unpredictable because its structural relation to the input is not intelligible. Nevertheless, when run more than once, it seems rather repetitive and predictable as the range of variations seems rather limited. The degree of authorship from the part of the user depends on his understanding of the process that leads from the input to the output.

In the case of programming, the structure of input depends on the user's knowledge; programming languages are very structured but one needs to know their structure in order not to be confused. Unpredictability lies within limits, set by the structure of the program. For example, the script used for this research gives rather

unpredictable results because it is based on randomised functions. "

[Anna Laskari, architect]

"Self organizing programs can definitely aid the design process, especially when it comes to quantifiable problems. So far, design representation software aid only in depicting instances of the designer's concepts, which is important in its own respect. However, self organizing software can help the designer make decisions that are based on situations that can be computationally expressed. In order for them to be seamlessly incorporated into the design process, such programs should be designed with intuitive interfaces.

The authorship of any design will still belong to the designer- architect but only part of it, since the final design outcome will depend on the efficiency of the program. The part belonging to the architect will have more to do with the creative process of establishing concepts and the framework, where the program can be of aid. As it is difficult for a single program to be able to aid any design process, the architect's ability to understand and adapt programming code becomes rather important."

[Tassos Kanellos, architect]

"The creative process relies on one's imagination and understanding of the architectural problem per se. Nevertheless, in several cases, a program might offer diverse and stimulating results; thus, enhancing the designer's- architect's role."

[Maria Kerkidou, architect]

" 'Many ways to skin a cat', the choice of the way matters less than the choice of the 'cat'! Defining constraints and objectives (e.g. fitness functions for G.A.'s, solvers, self-organizers) is necessary and highly desirable. The shift between 'CAD' drawing tool and computers/scripts/programming as a design tool is a shift in thinking about what they can do- not how."

[Chris Leung, research engineer]

"In the case of a script, the programmer has a certain control on the outcome. He defines something and lays back and benefits from the computational response. Having that in mind, the designer has to be equipped with the ability to program as to produce the intriguing results afforded. The design can be certainly influenced by the user but in some cases it is restricted by the initial topology- geometry. Authorship is on the user in any case and he can be attached to the result. Further development is afforded if used as a sketch where the abilities of the designer- architect can be further promoted and come on the surface."

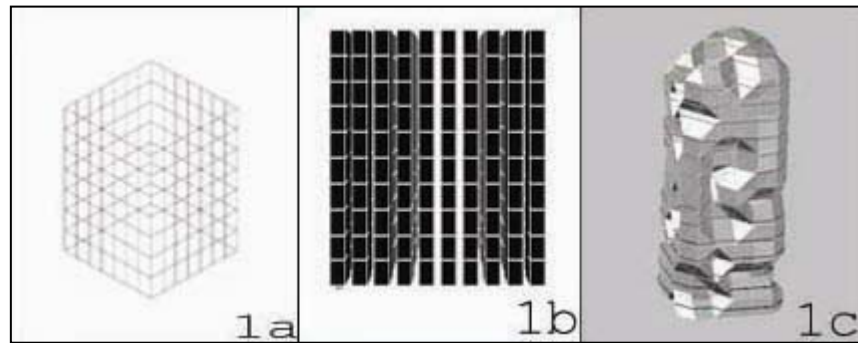
[Charis Charalampous, architect]



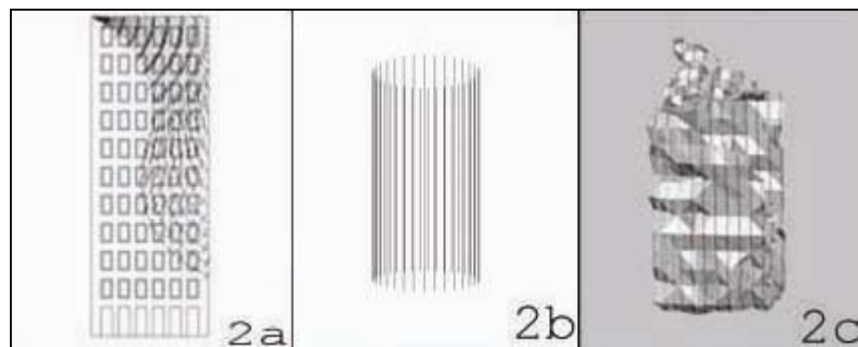
## Appendix II: Case Study results

---

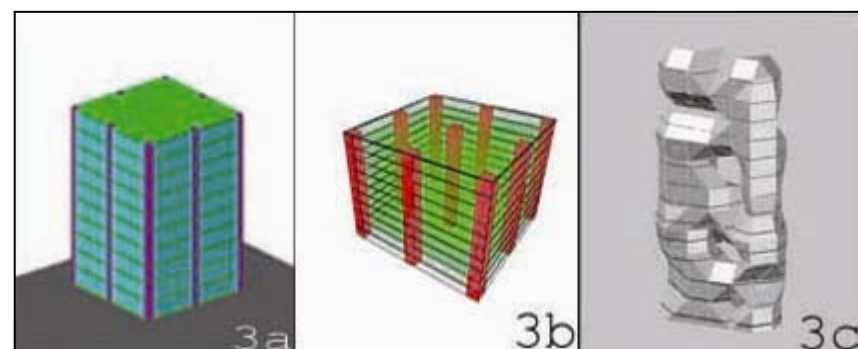
In this section the 17 case studies are set out, along with the type of program/programming tool used for the realization of each stage.



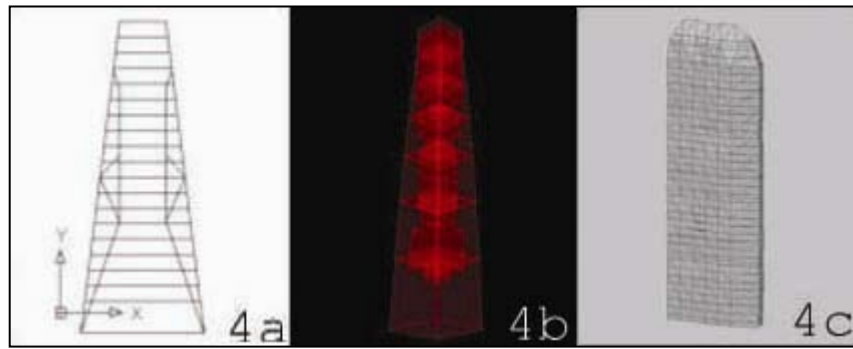
**Designer 1:** AutoCAD 2D | Processing | 'Faulty Towers'  
(Pilot sample)



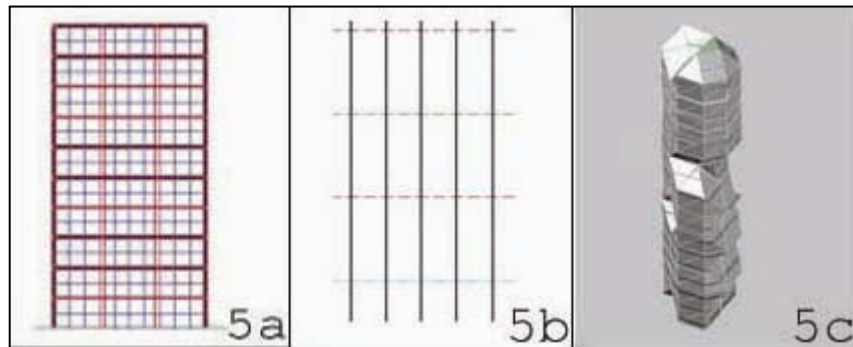
**Designer 2:** AutoCAD 2D | Processing | 'Faulty Towers'



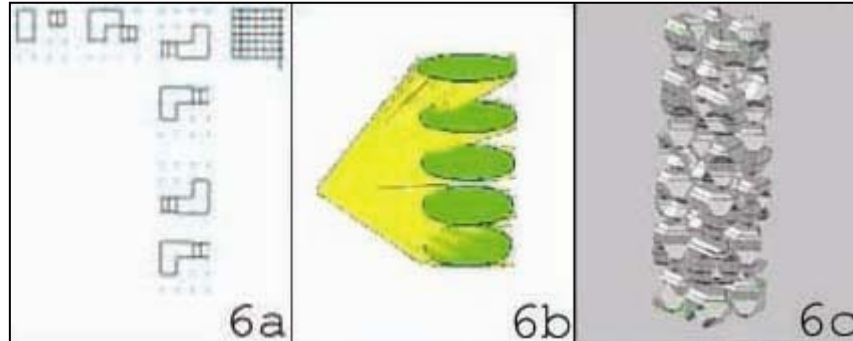
**Designer 3:** AutoCAD 3D | Processing | 'Faulty Towers'



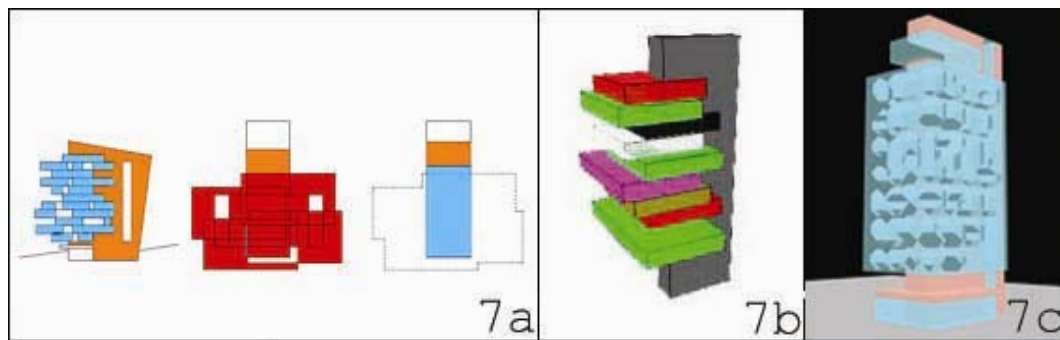
**Designer 4:** AutoCAD 2D | Processing | 'Faulty Towers'



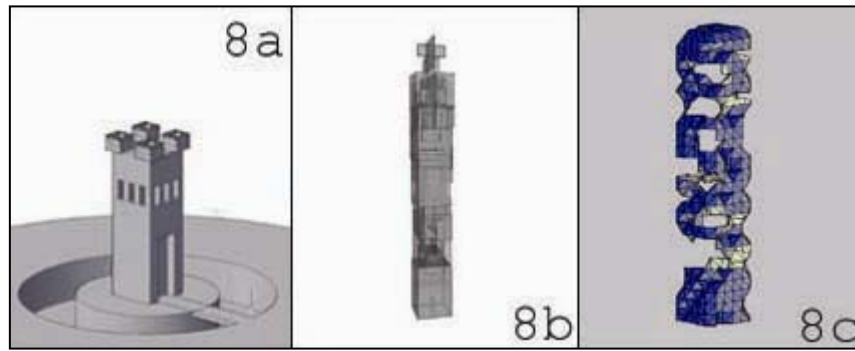
**Designer 5:** AutoCAD 2D | Matlab | 'Faulty Towers'



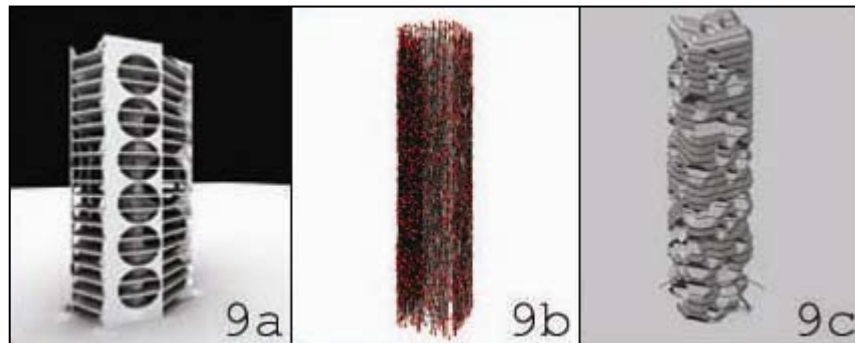
**Designer 6:** AutoCAD 2D | Processing | 'Faulty Towers'



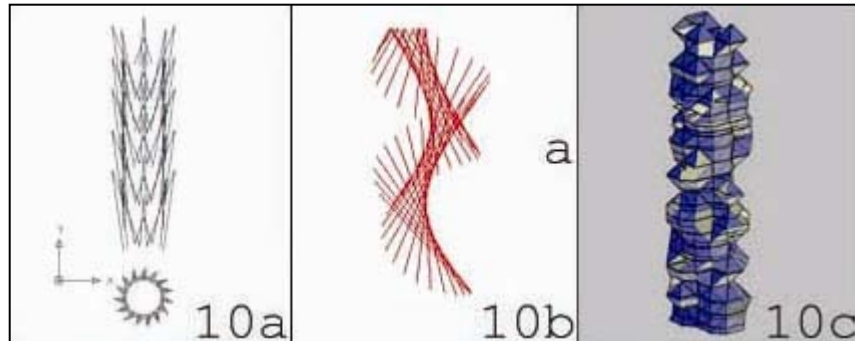
**Designer 7:** AutoCAD 2D | Processing | 'Faulty Towers'  
+3D Studio MAX



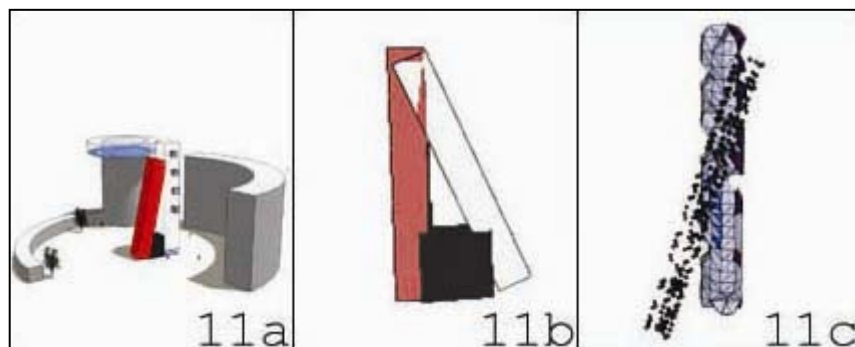
**Designer 8:** AutoCAD 2D+3D | Processing | 'Faulty Towers'



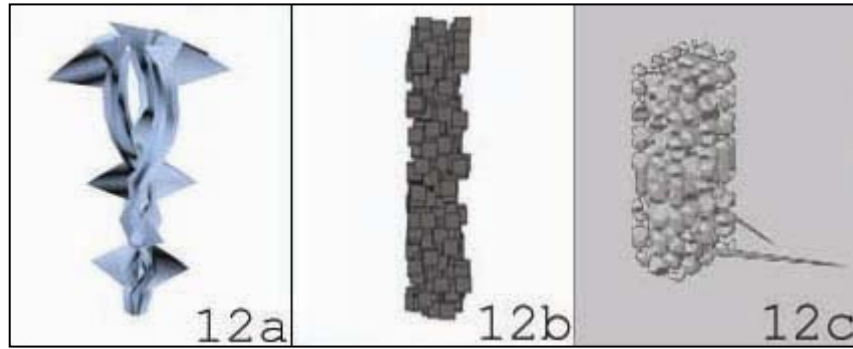
**Designer 9:** AutoCAD 2D+3D | Processing | 'Faulty Towers'



**Designer 10:** AutoCAD 2D | Processing | 'Faulty Towers'



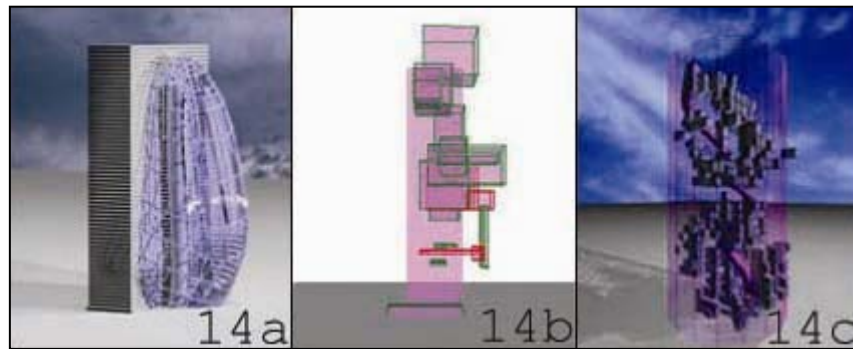
**Designer 11:** Google SketchUp | Processing | 'Faulty Towers'  
+Google SketchUp



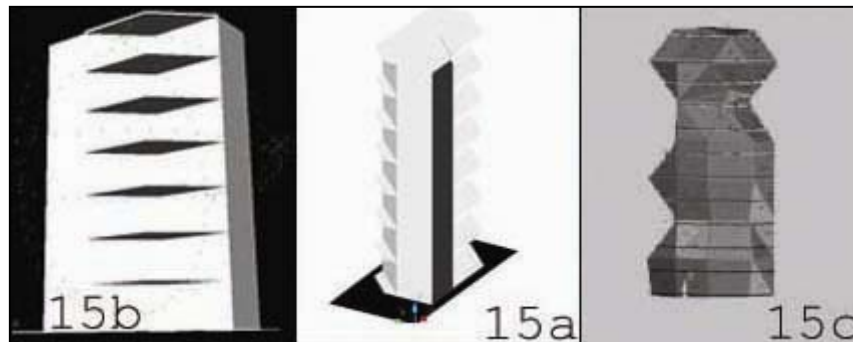
**Designer 12:** 3D Studio VIZ | Processing | 'Faulty Towers'



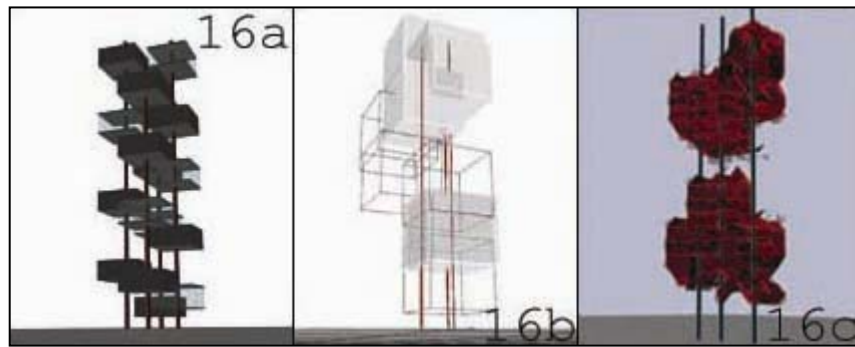
**Designer 13:** 3D Studio MAX | Processing | 'Faulty Towers'  
+Adobe Photoshop | +3D Studio MAX



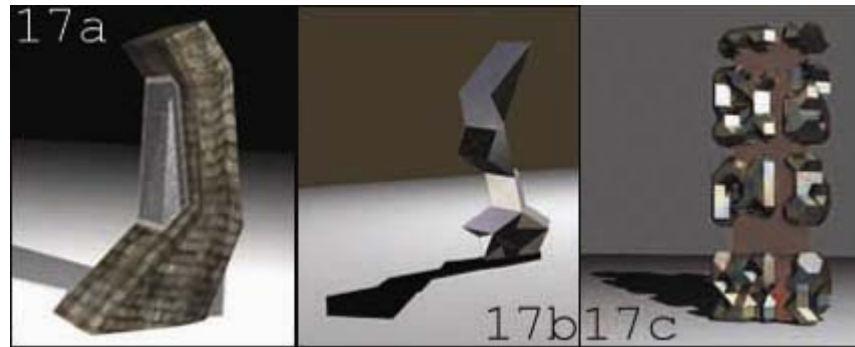
**Designer 14:** 3D Studio MAX | Processing | 'Faulty Towers'  
+3D Studio MAX



**Designer 15:** AutoCAD 2D-3D | AutoCAD VBA | 'Faulty Towers'



**Designer 16:** 3D Studio VIZ | Processing | 'Faulty Towers'  
+3D Studio VIZ



**Designer 17:** 3D Studio MAX | Processing | 'Faulty Towers'  
+3D Studio Max | 3D Studio MAX