Parametric Design as Digital Clay

How Thinking with Algorithms, Simulations, Repetitions,
Reactions and Random can Facilitate Learning

Avital BenEzra

Abstract

Parametric design is the practice of creating visual output by writing computer code. Exploring the different aspects of parametric design and the skills it requires and can help develop, this thesis suggests that using parametric design platforms is an opportunity for learning. By discussing central parametric design themes in relation to constructivist learning theories, I present the exciting learning opportunities that designing by code can offer. Analyzing parametrically designed projects, this thesis will focus on discussing the actual action of writing code. Concentrating on the process, I argue that great learning potential lay in using the inherent programming characteristics as an advantage rather then a setback.

Chapter 01
Introduction

An example for the learning potential that parametric design platforms hold can be initially illustrated through the following anecdote. Learning to visually program with parametric design platforms, the first lesson often includes a simple task: Designing a common geometric form, such as a triangle. This task provides an insight to a fundamental aspect of designing parametrically: It requires the student to explain what a triangle is. In order for the computer to visualize a triangle, the code would have to explain it: Draw a line between point A and B. Start another line drawing from point B to C. The end point, C, is the starting point for another diagonal line, which ends at yet another point. That point should be the starting point of the final line, which should find the shortest path to the first point. A triangle is a familiar shape that comes to mind easily. But thinking of it as a group of three lines, one point being one line's starting point and the other line's end point requires further engagement. Having to explain a triangle visually, allows for a new level of understanding. It is a physical, tactile comprehension even if the triangle appears nowhere else but on the computer screen. It is this kind of profound understanding that designing parametrically promotes.

There are several platforms (Resnick 2005) that help simplify the process of code writing by using pre-programmed backgrounds enabling user-friendly interfaces. These platforms are essential and extremely successful in encouraging children and inexperienced adults to engage with programming. They aim at creating accessible programing platforms to promote digital fluency and learning. Their emphasis is placed on the potential that coding can offer, in creating imaginative and creative outcomes such as animations, films, interactions and more.

This thesis is different in that it aims to look closely at the learning experience achieved by the process of direct coding rather than the mediated form of interaction with pre-programmed platforms. I suggest thinking about the exposed essence of programming: algorithms, simulations, repetitions, connections and randomness. By analyzing themes that are inherent to the process, rather than the designed solutions, this thesis explores how designing by code encourages new ways of thinking. Investigating parametrically designed examples, I will expose the process that enabled their creation and discuss the ways in which this process promotes learning.

The thesis is arranged into two chapter groups. The first is introductory to the field of debate. The second explores five different parametric design themes, analyzing their link to learning theories through examples. The first chapters serve as an introduction to the area of research and the main educational theories that will be further discussed in the following chapters. This first part will begin with an introduction to the field of parametric design, a short exploration of the history of the practice and its practical manifestation and uses today. The aim of this chapter is to define the term parametric design in the manner in which it will be referred to throughout this thesis. The following chapter introduces the relevant theoretical background. They present the main aspects of constructivist education, which is this thesis's starting point for the relationship between learning theories and parametric design. Later developments in relation to computers such as Seymour Papert's theories of the face of education in the age of computers, as well as Sherry Turkle's *Things to think with,* are the foundations on which this thesis is built. Researches such as Mitchel Resnick, who builds on Papert's theories and works on implementing them into real-life projects, will be discussed in relation to learning through coding. The work of Dale Dougherty and Stuart Brown is instrumental as well in thinking about experimenting, playing and making in order to learn. This chapter will outline the above theories in order to facilitate the detailed discussion, which will take place within the thematic chapters.

The second group of chapters will discuss different aspects of parametric design. Each of the five chapters will explore a unique parametric theme, which will be discussed in relation to relating learning theories and demonstrated by examples. The five themes are: Algorithms, Simulations, Repetitions, Conditions and Random.

The thematic chapters will discuss the process of parametric design as a fertile platform for investigation, and will demonstrate in detail the ways in which writing code encourages learning. The examples discussed vary in their purpose, chosen audience and form of execution. Some of them are printed, some are projected, and others are made of wood. However, they are all designed using code, and they share the ways of thinking that follow. The themes intertwine and many projects can be discussed in relation to more than one theme. Similarly, many of the learning concepts discussed relate to more than one theme. The division into chapters aims at formulating the information, and does not mean to suggest a rigid division.

In her 2007 book “Evocative Objects: Things we Think With”, Sherry Turkle writes: “We tie a knot and find ourselves in partnership with string in our exploration of space”. (Turkle 2007). I aim at discussing parametric design platforms as that string. In partnership with them, by engaging directly with their process, parametric design tools manifest as an exciting and highly rewarding learning platform.

Chapter 02
Parametric Design

This chapter defines the term Parametric Design in the context in which it will be used throughout this thesis. It includes a brief exploration of the field, its development through the years and the main contexts and uses in which it is most commonly used today. This discussion is providing context for the parametrically designed examples discussed in later chapters.

Parametric design can be defined as a design method in which the visual output is created by a set of variables and algorithms. It is a method that can generate numerous visual options based on a set of rules, defining relations and hierarchy, chosen and written by the designer. This set of instructions will generally be referred to as code, and will produce a visual output. Parametric design thus defines a hybrid of shape and code: visual output created with algorithmic input. Artists, musicians and designers have been experimenting with the use of code and themes that relate to coding in their work for many years, in many different ways. (Casey et al. 2010). Artists and musicians such as John Cage, Allan Kaprow, Sol LeWitt, Yoko Ono and others have used the concept of code for creating process-based conceptual art. LeWitt for example wrote ideas for drawings as lists of instructions, which had to be followed to produce the artwork.
This thesis will concentrate on projects that were created specifically using computer code, rather than more conceptual forms of code writing. Programming systems for the creation of visual images were starting to be engineered and developed in the 1950's and 1960's (Bohnacker et al. 2012, 32). These developments have been especially accelerated in the last decade, with many new parametric design platforms developed and a growing number of contemporary artists

and designers who incorporate the use of code in their work. There is a growing number of platforms available, at different levels of complexity, providing more accessible opportunities. Computer code is used to aid and enhance design outcomes and process in many different ways and degrees in all design fields, from architecture through interior, product, industrial, cinematic and graphic design. The architect and writer Patrik Schumacher coined the term 'Parametricism' and writes about it as "The great new style after modernism. Postmodernism and De-constructivism have been transitional episodes that ushered in this new, long wave of research and innovation". (Schumacher, 2012).

The practice of designing parametrically has many uses and manifestations, ranging from the practical to the poetic through everything in between. In terms of efficiency, parametric design systems are instrumental in finding cost, time and space effective solutions. They are often used extremely effectively for creating complex compositions or structures by automating processes. Accelerating calculations, they are able to optimise the use of resources such as time and materials. Parametric platforms can be used for producing design solutions considering restrictions and limitations, such as minimal waste of material for example. The ability to generate large amounts of variations in minimal time, while considering constraints, makes these platforms extremely productive tools. Parametric design platforms are also very useful tools for simulation and representation. By sorting and layering data, they are able to turn extensive amounts of information into coherent representations, which can be read and understood. As a result, these platforms are widely used for analysing crowd-sourced information. (Casey et al. 2010, 34-36). Able to act randomly and produce numerous variations, parametric design platforms can also be used for enhancing design development, becoming part of the creative process. Using visual parametric platforms is a mean for including a computational participant in the creation process, whether as a conceptual decision, as a result of practical needs, or both.

This thesis focuses on parametric visual systems that are used in order to enhance design processes, and not necessarily to optimize conditions. It explores the nature of the process these platforms share, and the ways of thinking they can promote. It does not explore the variations in interface and specific abilities between the different platforms.
The following chapters present examples for parametrically designed projects. The examples vary in terms of the specialized design field they come from, their context and uses. Nonetheless, all of the discussed projects have in common the use of code central to their process. The chosen examples are divided into five chapters, each concentrating on a specific parametric theme. Each theme represents a different attribute of parametric design. The thematic chapters explore

the territory and manifestation of each theme, proceeding to discuss its specific relation to theories of learning.

Chapter 03
Constructivism, Computers and Design

This chapter deals with introducing the educational theory that forms the base for this thesis. It serves as an introduction for the parametrically themed chapters, in which these educational theories will be discussed further, in relation to several parametric design projects.

Constructivism argues that people learn most effectively by constructing their own understanding and meaning of the world through reflecting on personal experiences. Seymour Papert is an MIT professor and the creator of the logo programming language. In his 1991 book 'The Children's Machine', Papert explains that the term constructivism commonly refers back to Jean Piaget's doctrine that knowledge cannot simply be conveyed to another person. Even when information is transmitted by telling it, in order to fully understand it, the listener will be reconstructing a personal version of that information. (Papert 1993, 142-3).
In his 1966 book The Savage Mind Lévi-Strauss uses the term *Bricolage* to refer to the knowledge-construction process he observed in primitive societies. Piaget has made similar observations in analyzing the behavior of children. Papert explains Bricolage as a cognitive process of tinkering, which involves experimentation and exploration. He discusses it as a metaphor for a bag of assorted tools, allowing the *bricoleur* to improvise, using what the bag can offer, in order to fix and improve mental constructions. Papert adopts the term *bricolage* from Lévi-Strauss, but claims it to not be limited to certain cultural groups or development stages. He writes regarding Lévi-Strauss and Piaget: "They failed to recognize that the concrete thinking they had discovered was not confined to the under-developed – neither to Lévi-

Strauss's "underdeveloped" societies nor to Piaget's not yet "developed" children. Children do it, people in Pacific and African villages do it, and so do the most sophisticated people in Paris or Geneva". (Papert 1993, 151). Constructivist learning theories encourage engaging in a mental Bricolage. Rather then concentrating on conveying specific information, Papert refers to constructivism as built on the assumption that learning is promoted by providing context for developing skills. Creating a fishing experience rather than providing fish. (Papert 1993, 138). Both Papert and Turkle reject the idea that *bricolage* is a stage in progression to a superior form of thinking, and that concrete thought is an early stage development leading abstract thought. They rather see *bricolage* as one of multiple styles of thinking and ways of knowing (Turkle et al. 1990, 141). They write about the positive affect that the use of physical models have on learning processes. In gears of my childhood, the introduction essay to his 1980 book "Mindstorms", Papert claims that any subject can be easy to learn if it can be assimilated into a personal collection of models. What and how an individual can learn, depends on what models he has available. (Papert 1980, xix). Following Papert and Turkle, many support their claim that the computer effectively provides context for creating such personal models, allowing for concrete interactions. Being able to simulate and take on numerous functions and offer endless possibilities, it is an especially powerful tool for engaging in multiple styles of learning, ways of thinking, and levels of understanding. (Papert 1980, Resnik 2002, Turkle et al. 1990)

The aim of this thesis is to suggest parametric design platforms as a step forward in using computers for thinking and learning. Combining programming with a visual approach, these platforms are especially effective in enabling ways of thinking creatively. Examining parametric themes, this thesis concentrates on the process of programming visually, rather than the outcomes it can produce.

Design is a practice that often presents opportunities for concrete thinking, tinkering, experimentation with materials and playful exploration of possible solutions. (Resnick et al. 1996, 130). Similarly to programming, it can offer a "conversation with the material" (Schoen 1983). In that aspect designing is similar to programming. The combination of both, in parametric design platforms is especially powerful, in that it expands and multiplies ways of knowing and levels of engagement. By concentrating on the basic process of designing parametrically, I claim that the specific outcome is of minor importance. The learning potential these platforms offer is widely spread, crossing disciplines, fields of knowledge and contexts.

03.1 **Parametric Design and learning**

Central parametric design attributes strongly relate to fundamental constructivist objectives. First, parametric design platforms provide the possibility of re-constructing learned information in a personalized manner. Demonstrated in detail in the thematic chapters, personal engagement, exploration and reassembling of information are deeply rooted in the process of designing parametrically. In addition, parametric design platforms, with their immediate visual and interactive output, are successful in consolidating abstract notions to concrete engagements.

Papert discusses the educational benefits of concrete thinking as opposed to abstract thinking. He describes the benefits of the shift towards concreteness from a constructionist point of view: "…The construction that takes place 'in the head' often happens especially felicitously when it is supported by construction of a more public sort 'in the world'… Part of what I mean by 'in the world' is that the product can be shown, discussed, examined, probed and admired. It is out there." (Papert 1993, 142). Parametric design platforms do exactly that – they consolidate an abstract idea into an interactive visualization – they 'put it out there'.

Further more, these platforms blur the line between doing and learning, offering a non-linear learning experience that is a combination of both. With the stages of research, experimentation, testing, finalization, and production all taking place simultaneously, parametric design is the ultimate tool for tinkerability: A stage for a playful learning and creating process. It is a process that fosters learning by putting emphasis on engaging with problems rather than producing solutions. (Papert 1993, 87). Finally, this thesis suggests thinking about parametric design as 'digital clay': a platform for creating what Sherry Turkle refers to as 'Objects to think with' (Turkle 2007) and Papert refers to as gears (Papert 1980). Objects that one can engage with and explore. In his essay 'Lifelong Kindergarten', Mitchel Resnick writes about the importance of making to the process of learning: "When you make something in the world, it becomes an external representation of ideas in your head. It enables you to play with your ideas and to gain a better understanding of the possibilities and limitations of your ideas" (Resnick 2013, 51). I believe that these ideas about learning are manifested especially powerfully by engaging with parametric design platforms. Throughout this thesis I suggest thinking about parametric design as a flexible, interactive tool for turning abstract ideas into concrete reality, enhancing the learning process in various ways. The following thematic chapters further illustrate how this comes about.

The learning potential parametric design platforms hold, lays in their encouragement of creating through engagement, experimentation, playfulness and analysis. The aspect that differentiate these platforms from other programming environments, lays in their inherent visual nature. They provide context for exploring through visualization. Colors, shapes and compositions are inherent and key to the parametric design process. As a result, these platforms allow a playful exploration of concepts such as process, connections, separations, dependency and rhythm, from a visual perspective. Experimenting with ideas manifested through form, offers an engaging experience that is able to develop, enhance and encourage creative thinking as well as other skills.

Chapter 04
Algorithms

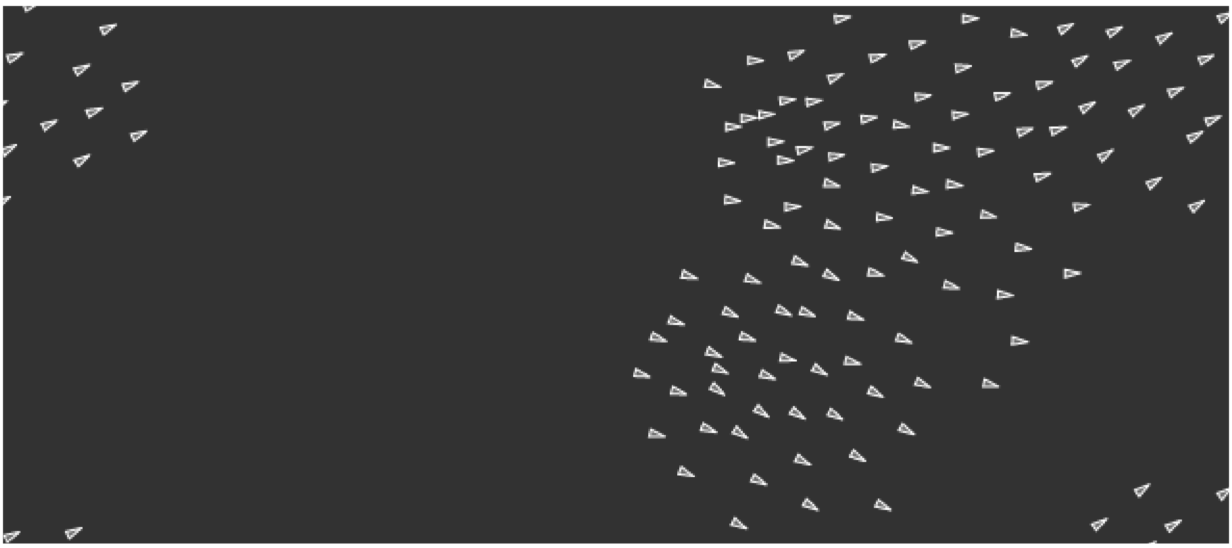
This chapter discusses the process of writing algorithms as a fertile platform for investigation. I will demonstrate the ways in which writing code encourages the search and discovery of the very essence of the process or subject in question. I will suggest that practice in concising complex ideas into their basic building bricks is a valuable lesson in exercising precision, selective judgment and in distinguishing between essential and incidental.

Algorithm is defined as "A process or set of rules to be followed in calculations or other problem-solving operations, esp. by a computer". (dictionary)
The term algorithm in this context can be further defined as a precise way of explaining guidelines for an action. (Reas 2010). It is a clearly written sentence in an agreed language, which comes to communicate an action to the computer. It is used as a mediator between the user and the computer; it is in fact a way of communication. Series of Algorithms written together create code. Algorithms are at the base of every parametrically designed platform. They are the ground on which code is built, and they enable all the other parametric themes, which will be discussed in the following chapters. The relation the act of writing algorithms has to learning manifests itself on two different levels. The first one relates to the specific subject of investigation, the other is general and applies to all. On a specific level, the act of writing algorithms requires an ability to analyze the subject in question, in order to be able to break it apart into a list of code written guidelines. Every idea must be thought of and stripped to its essence. The need to be explained to the computer also promotes a differentiation between essential and inessential

qualities of a subject. In a broader sense, the process of writing parameters demonstrates a shift from the abstract to the concrete (Papert 1993): It is transforming an abstract notion into an action. Code requires algorithms to describe a function and to trigger an action. Having to phrase an idea in 'physical' terms, is an exercise in concrete thinking, and can promote the development of concrete thought as skill. In parametric design platforms, the code in turn creates a concrete visualization. Thus the process of writing algorithms relates to concrete thinking both in the stage of building the code, and in the stage of interacting with it. This is one of the ways in which parametrically designed systems blur the line between doing and thinking. Both the process of planning and the process of creating occur simultaneously, and share characteristics. In 'gears of My Childhood' Papert writes about the benefits of learning through appropriate models, he refers to as gears. Describing his own personal experience as a child, using models in order to understand abstract mathematical ideas, he writes: "As well as connecting with the formal knowledge... (The gear) also connects with the body-knowledge... You can be the gear, you can understand how it turns by projecting yourself into its place and turning with it. It is this double relationship - both abstract and sensory – that gives the gear the power to carry powerful mathematics into the mind". (Papert 1980, xxi). Parametrically designed projects, as well as the process of constructing them by writing algorithms, are processes that echo the use of gears. It is a process of creating a concrete visualization of an abstract idea, a double relationship of both ways of thinking.

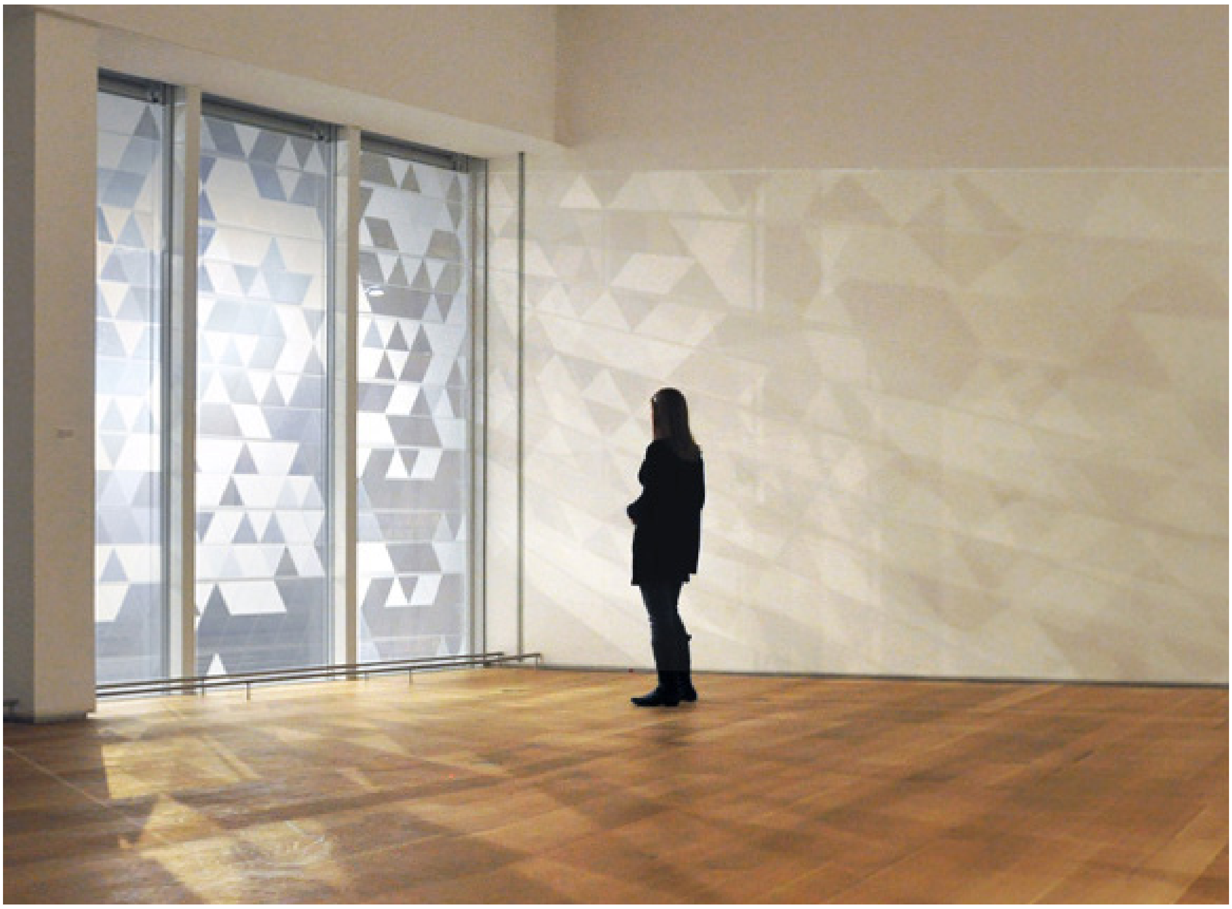
The following examples of parametrically designed projects suggest considering both the possibilities and the constraints writing algorithms presents, as promoters for learning and for developing valuable life skills.

04.1 Flocking (2011) Daniel Shiffman



This program simulates the flocking behaviour of birds. Each triangle steers itself in accordance with the others, mimicking a flocking pattern that is both structured and dynamic. A click on the mouse adds another triangle 'bird' to the flock, immediately adjusting to the rhythm and form of the other birds. Creating a parametric visualization of flocking provides an opportunity to explore the phenomenon. An investigation into the rules of flocking, in order to recreate them parametrically, promotes a deep understanding of the process. In order to create a flocking effect, rules of avoidance, alignment, and coherence must be articulated. Each 'bird' moves independently, steering towards alignment with the other birds, while maintaining separation by avoiding collision. (Shiffman 2012, 302). Creating a parametric visualization of flocking requires a profound understanding of the forces that influence this system. Once such a system is visualized, it exists in the world: it becomes concrete, it becomes "an object to think with". (Turkle 2007). The opportunity of thinking with a flocking visualization, allows reconstruction of the rules of the system, breaking it apart and putting it back together by changing parameters. In addition, it can promote understanding of similar behaviors and phenomena such as a school of fish, a swarm of insects, and herd behavior of land animals. (Shiffman 2012). It is the kind of learning that encourages further learning.

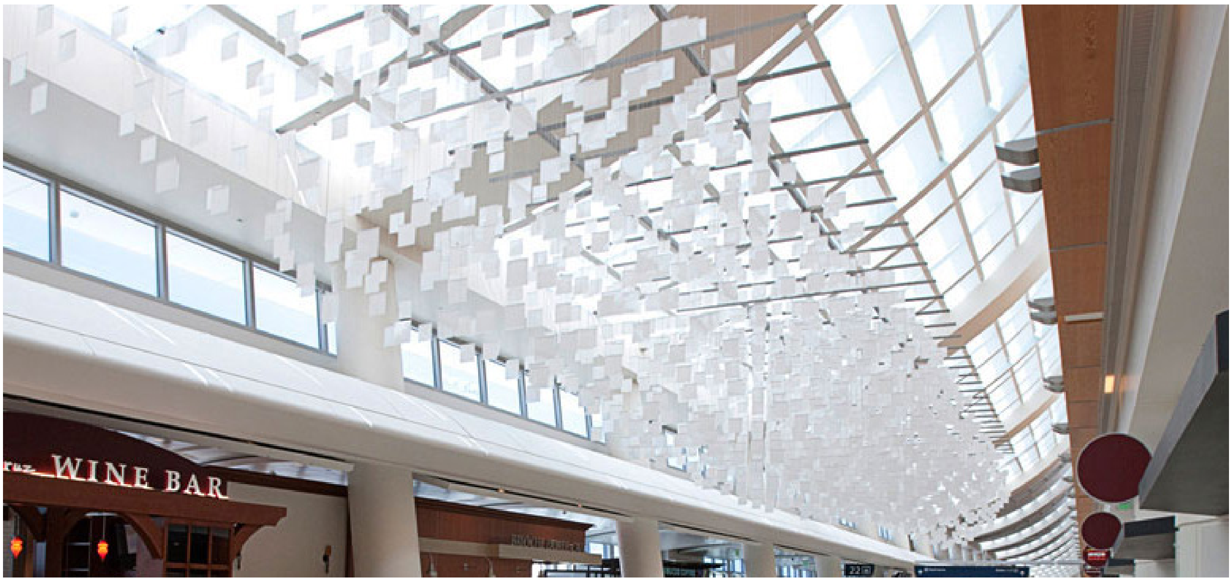
04.2 Shade (2011) Simon Heijdens

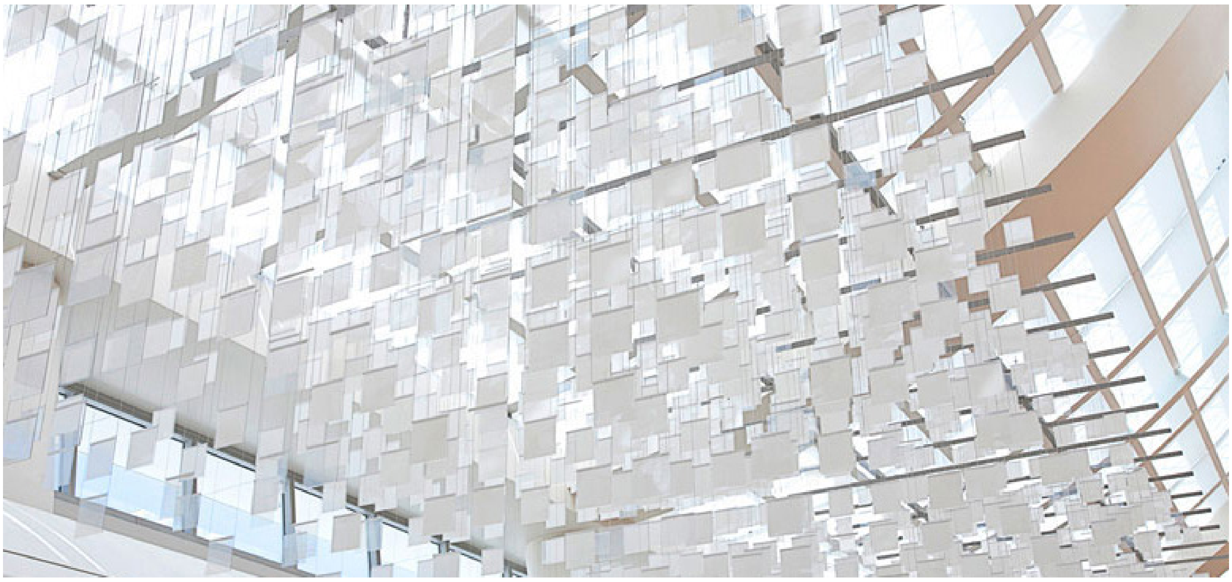


Shade is an installation projected on a wall by a set of screens physically applied to an adjacent glass window. It is a dynamic, constantly changing wallpaper, mirroring the outdoor condition in the indoor space. As explained in Heijdens's website: "A responsive skin to the windows of a building that filters daylight into a moving projection of shadows, that translates the ever-changing natural timeline of the outdoor to the static and perpetual indoor space..."
Through variably blocking or passing light, shade transforms sunlight into a spacial, graphically projected moving image of shadows. Its surface holds a grid of triangular cells that are each individually able to change their level of opacity, hence block or pass light. The graphic shadows projected on the floor, walls and ceiling of the space - varying from sharp and graphic, to fade and blur depending on the weather and time of day - reveal the geometrical patterns of wind that pass the building on the other side of the glass. The pattern of movement is choreographed by the measurements of an outdoor sensor. As both the angle of the sun and the patterns of wind are continuously changing throughout the day and year, the space becomes reconnected with an evolving, unplanned and dynamic natural timeline.
(www.simonheijdens.com). Shade is a responsive program that mimics the outdoors conditions by translating live information into pre-determined behavior, as outlined in the algorithms that compile it. Parametric design platforms promote this exact manner of thinking. By writing a set of parameters, an abstract notion becomes concrete. It allows learning skills by actual engagement with real context. Further more, In order to write the algorithmic program for Shade, several crucial issues had to be explored and thoroughly understood. First, in order for the wallpaper to simulate the reality outside, the very essence of light and shade had to be analyzed. In order to turn light into a graphic representation, the behavior and quality of sunlight must be explored. Decisions had to be made, which were not only crucial for the programing of this artwork, but that reveled facts about a physical artifact. Such as what is the minimal amount of light required to create a visibly noticeable effect? How does the different times of day influence the amount and saturation of light outside? What would be the ideal differentiation between each individual panel in terms of their reaction to the sunlight, in order to create a distinctive pattern? These questions and many more were addressed in the creation of this installation, promoted by the challenges algorithm writing presents.

The act of writing parameters holds the potential of offering a powerful learning experience both by encouraging the development of skills and by offering a concrete and relevant context for the skill to be acquired.

04.3 eCloud (2010) Dan Goods, Nik Hafermaas and Aaron Koblin





“The eCloud is a dynamic sculpture inspired by the volume and behavior of an idealized cloud. Made from unique polycarbonate tiles that can fade between transparent and opaque states, its patterns are transformed periodically by real time weather from around the world”. (www.ecloudproject.com). This installation simulates reality: It transfers information into a concrete visualization. Similarly to Shade, eCloud’s algorithms reflect a profound understanding of a behavior of a cloud. The program that runs eCloud can decode live weather information into an accurate simulation: It can behave like a cloud. An understanding of the metrology phenomena and their effect on weather conditions is integrated in eCloud. Each individual panel reflects this understanding: Each occurrence in the cloud reflects an analyzed and explained cause. eCloud is a profound lesson in metrology, its creation allowed and demanded a deep understanding of the creation and behavior of clouds. All these learning benefits are further complimented and intensified by the concrete nature of the parameterizing action. It proposes actually creating a cloud rather than only discussing one.

04.4 Lightweeds (2006) Simon Heijdens

Lightweeds is a lively installation of large seaweed, projected onto walls, floors and celling. Seaweed is explained in Heijdens's website: "A living digital organism growing onto an indoor space, through which the space regains the natural timeline that it has walled out. Uniquely generated plant families that grow up, move and behave closely depending on actual sunshine, rainfall and wind as measured live outside. On passing human traffic they bend, loose their seeds and pollenate to other walls throughout the space, to make up a constantly evolving bio system that reveals the character of the space and how it is used". (www.simonheijdens.com)

"Lightweeds" is a digitally alive plant that reacts to real condition by mimicking real plants reactions. It moves and grows as a planted plant would: It reflects an understanding of the actual biological conditions, causes and effects of real plants.

It is an addition to the knowledge this installation both requires and reflects, as explained in the previous examples, "Lightweeds" offers another level of reflection. The weeds do not only react to outside condition. They are also sensitive to movement of people within the space in which they are installed. Their behaviour is in fact a reflection of the pattern of human behaviour in the space, as much as it is of the outside conditions. "Lightweeds" reflects the usage of the space by using it as a catalyst for change. By doing that it turns yet another extremely abstract notion into a concrete, tangible representation.

In conclusion, the algorithmic aspect of parametric design is closely linked to notions of the benefits of concrete thinking. By the limits this practice presents, through the encouragement it offers to acquiring knowledge and the concrete nature of its artifacts, parametric design platforms have potential for teaching us more than code. It does not only encourage investigation into a given subject due to its requirements. More importantly, it makes us think physically, concretely, about subjects that are usually taught and learned only theoretically.

Chapter 05

Simulations

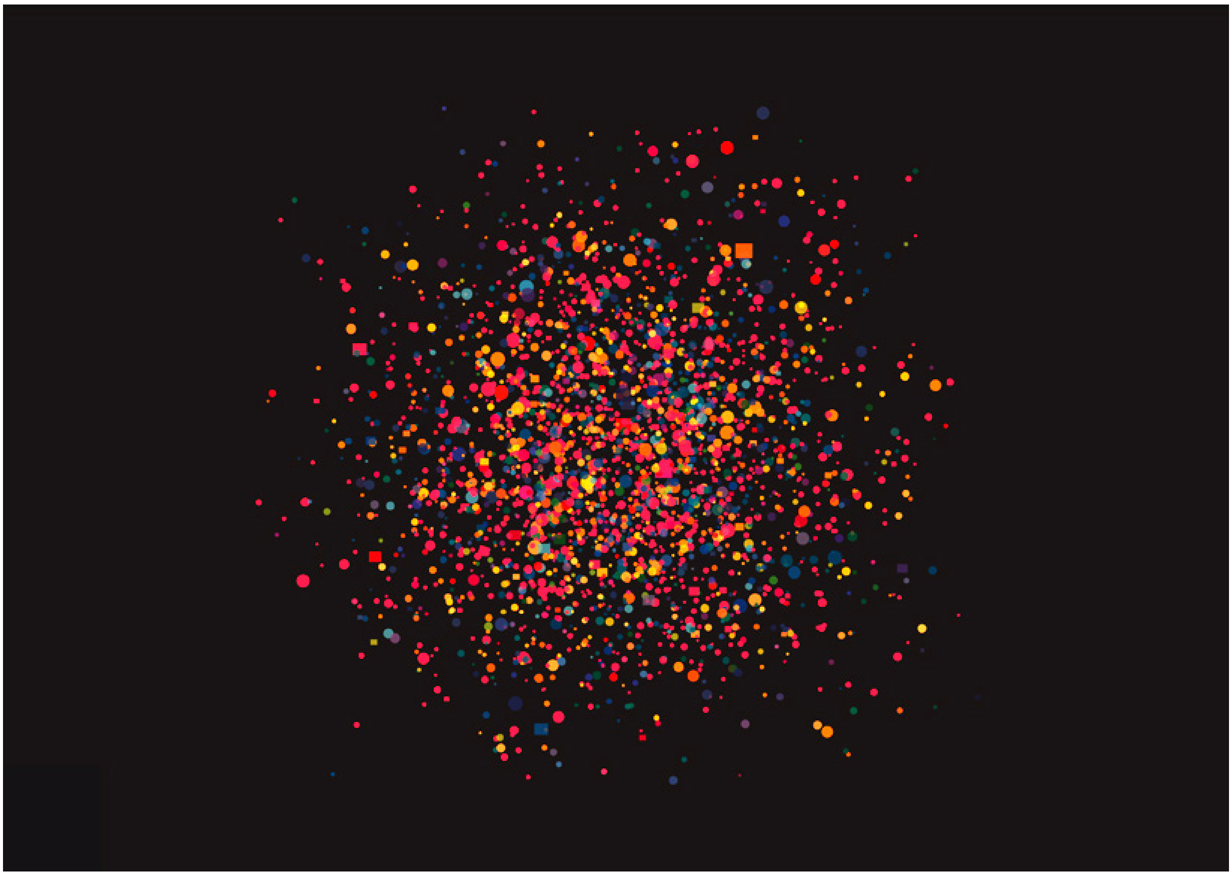
This chapter will explore the importance that re-construction of information has to the learning process. The discussed examples will demonstrate the ways in which personalized re-construction is encouraged through parametrically designed projects.
Computer and code are often used for creating simulations that are imitating, or predicting, real life. For example, the ability to simulate physics through 3D computer programs has had a substantial impact on architecture and engineering. (Reas 2010, 149). The type of simulations that will be discussed in this chapter are of a different nature. They are not mimicking, or meant to predict, real life situations. Rather, they aim at visualizing information that is otherwise formless. They exist in order to provide a tactile, solid existence to abstract notions. Additionally, the dynamic nature of parametrically designed platforms allows them to be flexible. They provide a concrete participatory experience of navigating through an abstract knowledge field. Casey Reas writes about this kind of simulation: "Simulation is the creation of the possibility of form". (Reas 2010, 149). The unpredictable, visionary nature of these simulations, differentiate them from more accurate, mirroring representations, such as the ones discussed in the Algorithms chapter.

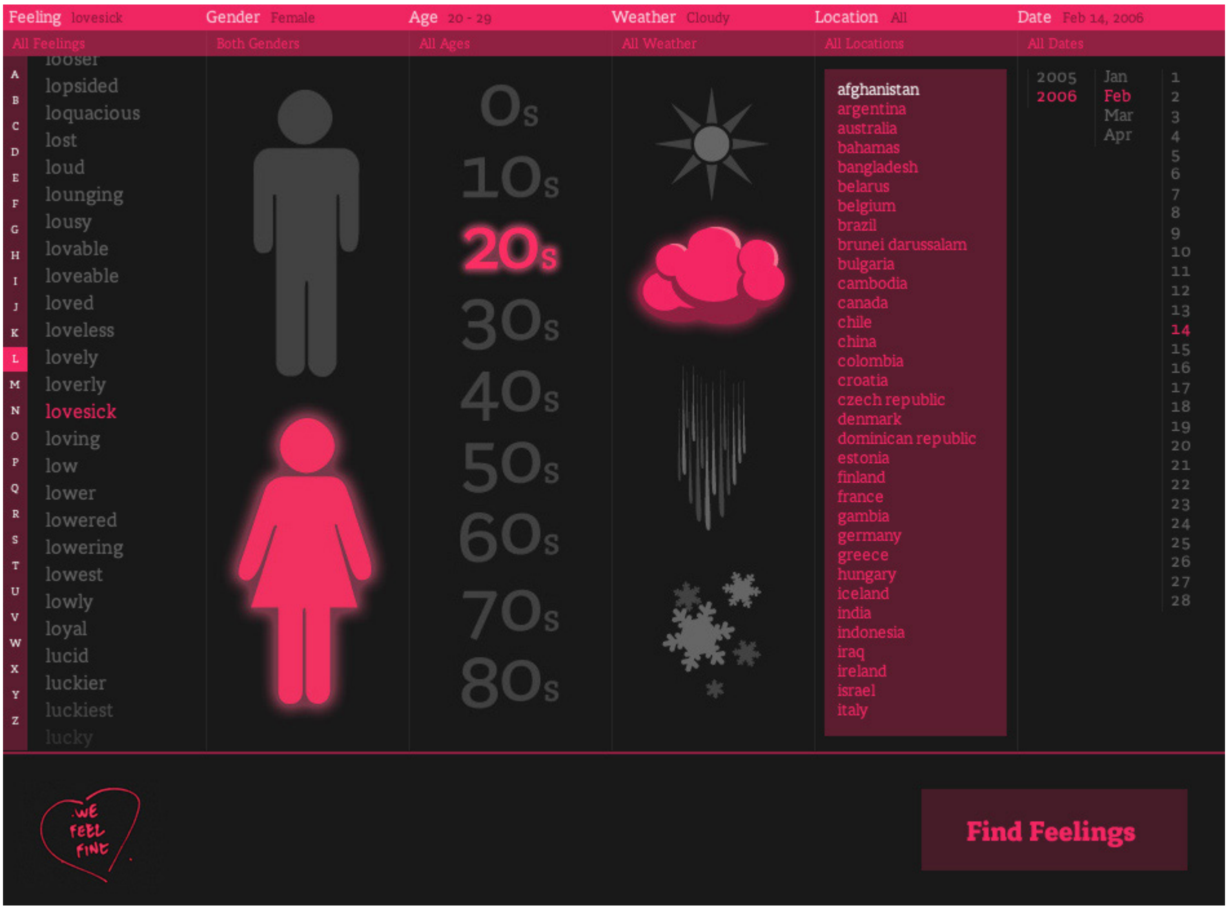
Visual parametric systems are often used for the display of multilayered, complex information. The highly interactive, adaptable and flexible nature of code generated design, makes it an ideal solution for making sense of extensive amounts of information through visualization. (Reas 2010, 147). Parametric designed systems are able to construct vast and complex networks of layered information, while allowing manipulation and selective navigation. These participation-enabling features are the focal point of this chapter. The following projects are both examples for dynamic systems, formalized to allow selection of a personalized point of view. In other words, they are all open platforms, allowing personalized navigation through information, promoting customized learning. Another thing the following examples have in common is the fact that they are all using information in order to create concrete visual representations.

Therefor, these parametrically designed examples echo Papert's theory in that they are neither abstract nor static. As he writes: "...My personal reconstruction of constructivism, has as its main feature the fact that it looks more closely than other educational –*isms* at the idea of mental construction. It attaches special importance to the role of constructions in the world as a support for those in the head..." (Papert 1993, 143).

There are two main ways in which simulation, as a parametric theme, is closely connected to learning. First, it can solidify naturally theoretical information into a visual representation. Second, it enables participation and experiment. These ideas will be discussed further through examining two parametrically designed examples.

05.1 We Feel Fine (2005) Jonathan Harris and Sep Kamvar



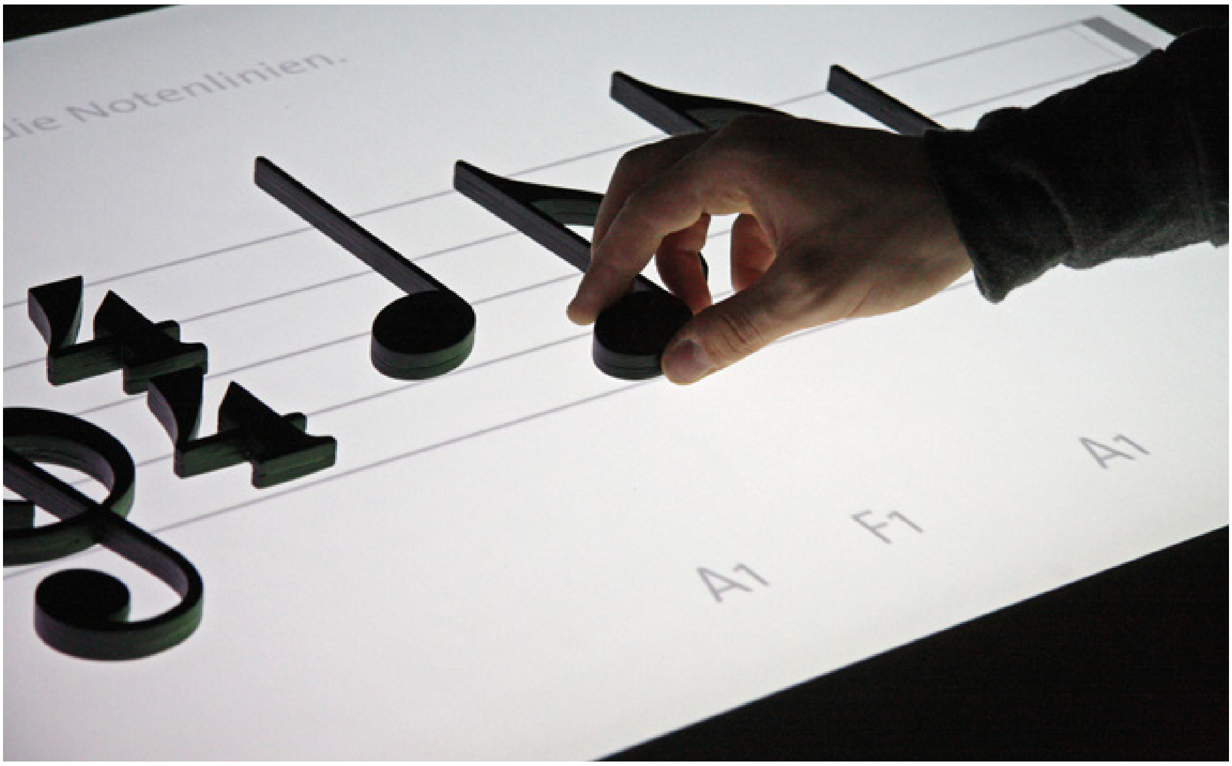


We feel fine is a graphic representation of automatically collected and assembled human emotions. It is an interactive parametric system allowing extensive exploration of the visually presented data. As explained by the artists on the project's website: "Since August 2005, We Feel Fine has been harvesting human feelings from a large number of weblogs. Every few minutes, the system searches the world's newly posted blog entries for occurrences of the phrases "I feel" and "I am feeling". When it finds such a phrase, it records the full sentence, up to the period, and identifies the "feeling" expressed in that sentence (e.g. sad, happy, depressed, etc.). Because blogs are structured in largely standard ways, the age, gender, and geographical location of the author can often be extracted and saved along with the sentence, as can the local weather conditions at the time the sentence was written" The result is a database of several million human feelings, increasing by 15,000 - 20,000 new feelings per day. (www.wefeelfine.org, 21.01.2014). Using a series of playful interfaces, the feelings can be searched and sorted across a number of demographic slices, offering responses to specific questions like: do Europeans feel sad more often than Americans? Do women feel fat more often than men? Does rainy weather affect how we feel? What are the most representative feelings of female New Yorkers in their 20s? What do people feel right now in Baghdad? What were people feeling on Valentine's Day? Which are the happiest cities in the world? The saddest? And so on... The creators of We Feel Fine claim that it is an artwork authored by everyone. It will grow and change as users and contributors grow and change, reflecting thoughts, feelings and states of mind. They conclude by stating: "We hope it makes the world seem a little smaller, and we hope it helps people see beauty in the everyday ups and downs of life". (www.wefeelfine.org, 21.01.2014).

We Feel Fine provides a fascinating exploration experience and insight into the emotional patterns that forms our collective online existence. Allowing a customized point of view enhances this experience further. The interface is arranged into a composition of individual particles, each representing an emotion. Variables such as color, size, shape or opacity reflect their characteristics. For example, bright colors are used for representing happy feelings, while darker shades represent sad ones. *We Feel Fine* offers numerous navigation options with six different perspectives to choose from. Users can navigate through the system by visual emotion representation, written text, graphs, most common feelings or most unusual ones. Each chosen angle then allows further customization of the exploration by choosing search criteria: It is possible to search each angle by various variables such as the kind of emotion, place, weather, age of person, geographical location, time of year and more. *We Feel Fine* is a relevant example for allowing re-construction of information and of comprehending through personal experience. It provides the viewer with a set of tools for constructing a personal meaning from a large assembly of data. It is an opportunity to engage in a *bricolage* type of activity. As Papert writes: "I use the concept of *bricolage* to serve as a source of ideas and models for improving the skill of making – and fixing and improving – mental constructions". (Papert 1993, 144). It does not simply convey information but rather it allows each user to create their personal meaning. The ability to personalize information is another promoter to the learning process. It offers an opportunity to construct information according to personal preferences and interest. We know that people learn most when engaging with a personally meaningful project. (Resnik 2012, 43).

The next project uses a different media and approach, and conveys a different type of information, and yet it is similar in that it allows personal reconstruction of meaning.

05.02 Noteput (2009) Jonas Friedemann Heuer and Jürgen Graef



Noteput is an interactive musical table. It includes a notepad surface and a set of tangible musical notes, allowing playing of music in an innovative manner. Noteput is described in the book “A Touch of Code: interactive installations and experiences”: “All basic keys, note values and accidentals exist as single wood elements. Whole, half, quarter and eighth notes differ not only in their form, but also in their weight: Long note values are heavier than short ones. The table has two modes: A standard mode, where you can place notes on the table in a playful and experimental way and explore the related music outcome. An exercise mode, where exercises and tutorials sort by topic and difficulty have to be mastered. To activate “Noteput”, one simply has to put the treble clef on the table. As soon as a note is placed on the staves, the respective sound is heard. That serves as preview and orientation while placing the notes. If several notes are on the table, one can hit the play button and listen to the notes in relation to each other while considering note values. In addition to piano other instruments like guitar, flute, vibraphone or e-piano can be chosen”. (Reas 2010, 54). *Noteput* does not only visualize music as it is played, but it also provides a continuous visual existence to played notes. Musical experimentations physically exist, allowing the user to manipulate the notes while both hearing and seeing the consequences of his actions. The ability to choose and modify only one note within a composition promotes understanding of that note’s influence on the overall outcome. *Noteput* visualizes the musical information, turning the experience into a concrete, physical exploration. It is able to teach the logic of notation and music allowing personal reconstruction of information. Similarly to the previous example, the specific content is of a secondary importance in this context. The option of simulating visually abstract information, promotes better understanding. It is a musical *Bricolage* kit, allowing independent learning. Papert writes in criticism of abstract thinking: “The point of abstract thinking is to isolate – to abstract – a pure essential factor from the details of a concrete reality”.(Papert 1993, 149). *Noteput* does the exact opposite. It connects musical sounds with their physical, visual simulation. Adding a variant of physical weight to notes, to indicate sound length, is a clear example of that solidifying process. It becomes a thing to think with – the ‘objects that matter’ Sherry Turkle writes about. (Turkle 2007, 10). The fact that each note is independent, allows a customized exploration of the system, promoting endless options.

The two projects discussed above demonstrate the parametric simulation ability and the learning potential it carries. It allows conveying information while encouraging engagement and participation. It is a fertile ground for thorough exploration of any type of subject. It has the ability of transforming knowledge into a modular, customized system, thus enhancing and improving the comprehension experience. Understanding of any kind of information will be encouraged and enhanced by integrating personal options for exploring it.

Chapter 06

Repetitions

This chapter will discuss the ways in which repetition occurs, and can promote learning, in three parametrically designed projects. Repetition exists and is used for numerous ends in art, literature, poetry and more. The aim of this chapter is to explore the specific manifestation of repetition in visual parametric systems.

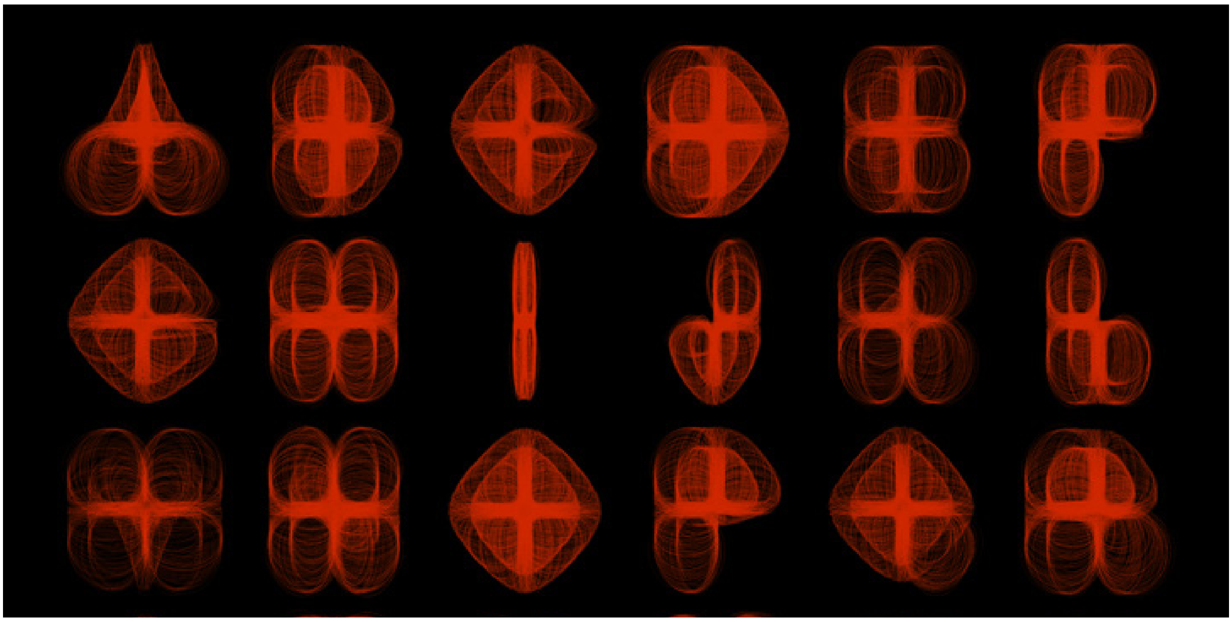
There is an obvious link between repetition and learning: we are used to repeating taught information to make sure we can remember it and use it in the future. However, the purpose of exploring parametric repetition in this context is rather to discuss its ability of teaching us about processes, rhythm and natural recursion.

Repetition is an essential and significant part of code, deeply embedded in its writing process. (Reas 2010, 45). The programming language processing for example, has repetition set as default for all actions: Unless otherwise instructed, the computer will repeat each command forever. The capability to control and manipulate an endless repetitive sequence is enabling development and change while maintaining a common ground. With repetition as constant, changing variables around it allows exploring range of possibilities within a chosen degree of similarity. Visual repetition can generate families, or generations, of visual elements. Rhythm, whether visual or musical, can be viewed as created by interrupting a repetitive cycle. A visual rhythm is the rule that determines visual repetitions such as patterns. (Reas 2010, 49).

The ability to interfere with a visual rhythm sequence is the ability to reconstruct the image.

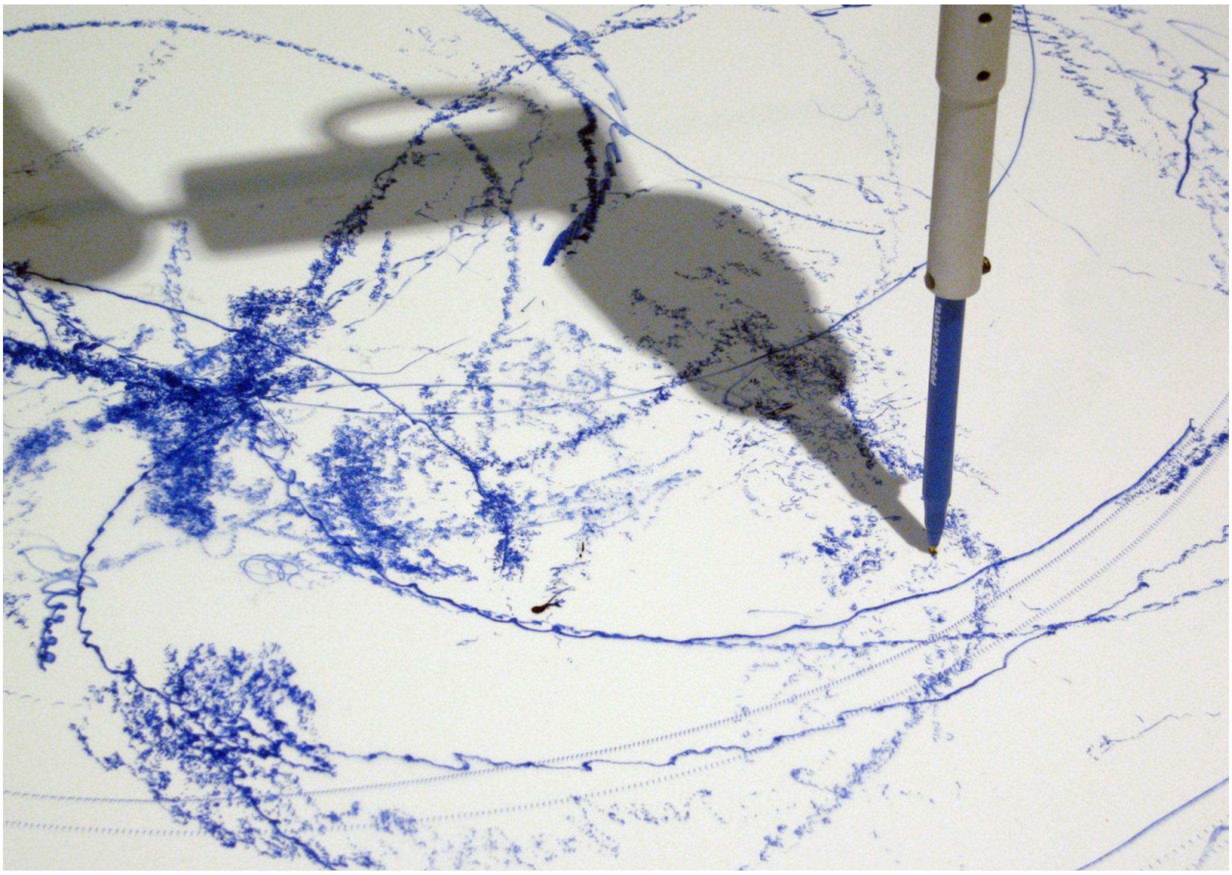
Automated repetitive action, combined with the ability to interfere its process, has the potential of creating endless visual options. Recursion is a closed repetitive technique. It is the process of repeating objects in a self-similar way; it is a function that includes a reference to itself as part of the function. (Reas 2010, 63). The computer code action that relates to these systems is called a loop, referring to its endless, repetitive nature. Fractals are an example for a recursive system. In the book "The Nature of Code" Daniel Shiffman quotes the mathematician Benoit Mandelbrot and defines fractals as: "a rough or fragmented geometric shape that can be split into parts, each of which is (at least approximately) a reduced-size copy of the whole". (Shiffman, 2012, 367) Repetitive systems created in visual parametric platforms, allow interfering and manipulating the visual sequence. They promote learning by offering a personalised exploration of repetitive processes, rather than a passive observation. This kind of participatory engagement with repetitive systems can teach us about natural recursive processes. It also allows isolation of a single element from a complex system. With the rest of the components constantly repeating, it allows detailed examination of a specific part. Almost like a digital laboratory, it can offer isolated conditions. These learning opportunities will be further discussed through the following parametrically designed examples.

06.1 Erotica A to Z (2007) Yeohyun Ahn



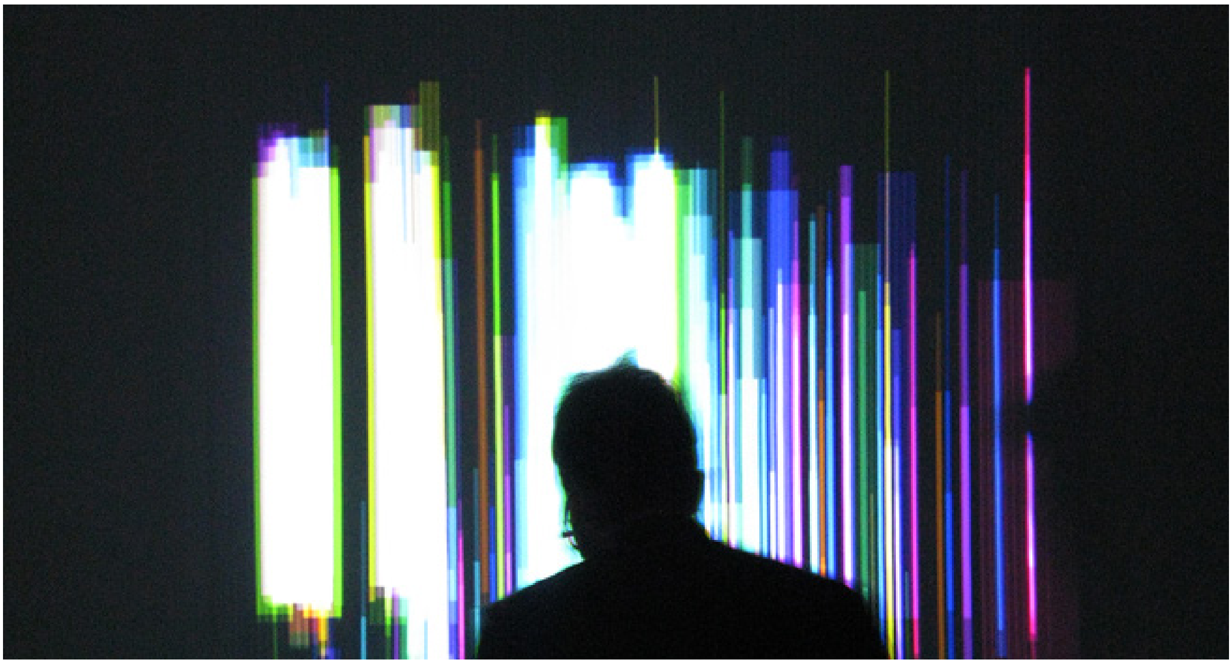
Erotica A to Z is a code-generated typeface. It is part of a series of ten typefaces created in the visual programming language Processing. Erotica, as other typefaces, is a family of letters: It uses the same building blocks to construct similar yet different components. They are all made with the same visual DNA, the variations creating a diverse typeface. Erotica is an example for manipulation of a repetitive system in order to explore visual options within a defined set of constants. It allows a visual investigation: If one component of the central repetitive pattern changes, how might it affect the individual letter? The relationship between a letter and the complete set, a detail and the whole, is also open for exploration. The open repetitive system promotes better understanding of it by encouraging exploration. The repetitive nature promotes ideas of variation and similarity, and the conditions in which the two might exist together. Exploring visual individualism from one side, and continuity from the other.

06.2 Drawing Machine 3.1415926 v.2 (2008) Fernando Orellana



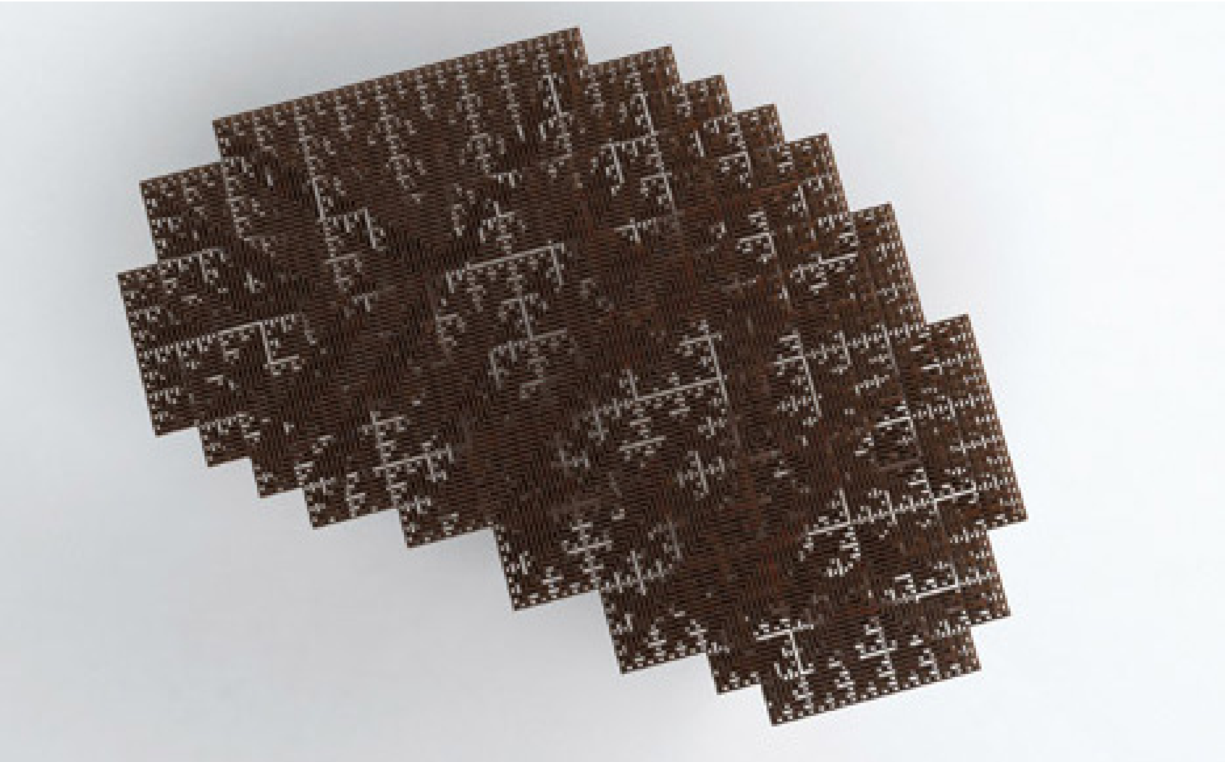
Drawing Machine 3.1415926 v.2 consists of a three-tiered mobile sculpture that is driven motor vibration, controlled in two ways. First, by the computer code dictating drawing instructions. Secondly, by monitoring one or two microphones, giving the machine the ability to "listen" to its environment. When it hears a loud enough sound it uses that information directly to create marks. In this way the machine collaborates with its environment; using both its program and what it hears to make drawings. (Reas 2010, 126). Drawing Machine combines constant repetitive functions with inconstant ones. The sounds in the environment interrupt the rhythm of the repetitive code and leave their mark on the paper. The Drawing Machine produces an account of the noise patterns around it in any given time. That information is registered thanks to its irregularity in comparison to the repetitive nature of the system. As it is static and uninterrupted, it provides confined points of reference for investigation.

06.3 Clavilux 2000 (2009) Jonas Friedemann Heuer



Clavilux 2000 is a platform for visualizing music. In *Clavilux 2000,* every note played on the keyboard creates a new line, which appears on a screen. The line follows in its dimensions, position and colour the way the particular key was stroke: The length and vertical position show the velocity, the stripe's width reflects the length of each note. The colours create an image that gives the viewer and listener an impression of the harmonic relations. Notes belonging to one specific tonality will get colors from one specific area of the color wheel. Therefore each key is translated to a unique color scheme. The more different tonalities a piece has, the more colorful the visualization will be. The visualization of *Clavilux 2000* does not disappear when the music fades away. All stripes stay and overlap each other in an additive way, so at the end a kind of pattern remains – a summary of the music. Since not only the notes played but also the interpretation of the piano player are influencing the outcome, visualizations for one single piece of music will vary. Therefore the stripes can reflect something the nature composition as well as the specific performance. It is also possible to play the notes in a loop. As a result it is possible to compare how the current note-sequence sounds like and how changes of the notation influence the music. (www.jonasheuer.de, 22.01.14). *Clavilux 2000* offers the user the option of playing a musical loop: a constant repetitive sequence. The loop enables examining individual notes in relation to the whole composition. It can reveal the influence every single note has on the composition as a whole. It provides an opportunity of interfering with the rhythm by adding or reducing from the loop. In other words, it allows a personal reconstruction of a musical harmony, enhancing the experience and promoting a deeper comprehension of the piece played.

06.4 Fractal Table (2008) WertelOberfell



The Fractal Table uses the logic and behavior of fractals for creating an artificial fractal system. The designers describe the work on their website: "Fractal table is a result of studies into fractal growth patterns that can be found in nature and which can be described with mathematical algorithms. (It) is a fragmented geometric shape that can be split into parts, each of which is... a reduced-size copy of the whole, a property called self-similarity. The fascination for us as designers lies in the objects' grown and organic nature but also in its structured and mathematical quality. Both in terms of size and complexity Fractal Table pushes the manufacturing process to its limits... Treelike stems grow into smaller branches until they get very dense towards the top to form a quasi-surface. The structure starts quite unorganized at the bottom and gets progressively organized till it ends in a regular grid, thus a progression from an approximate fractal to a fractal with exact self-similarity." (www.platform-net.com, 19.01.2014).

A fractal visual parametric system is open for manipulation and investigation. Its creation demands an understanding of the fractal structure, and it allows examining the influence of changing variables. This mental engagement with fractals can promote a better understanding of the natural phenomenon. Fractals exist in nature in various fascinating forms. Manifested in plants, mountains, seashells, snow flakes, crystals, vegetables, fruits, corals, fjords and more.

A digital fractal has the potential of being a digital learning model for the understanding of an external complex system. The repetitive nature of the fractal system allows exploration of the system as a whole, as well as promoting a profound understanding of each individual fracture, by analyzing its relation to the other elements.

The discussed examples explore the idea that repetition as a parametric trait has the potential of enhancing the learning experience, by providing options for reconstructing meaning by navigating through repetitive systems. It also can provide an insight into specific forms of repetition such as patterns, visual families and fractals. Multiplicity and reoccurrence have the advantage of exposing connections, relationships, influences, cause and effect. Interfering with a repetitive pattern can isolate and focus on details, reveling information.

Chapter 07
Conditions

This chapter discusses the dependent nature of parametrically designed systems.
It will explore the corollary nature of visual systems produced by code and the meaning and benefit of navigating through a consequential platform. The examples will demonstrate the ways in which the ability to affect a system entirely by changing only one variable, relates to constructivists educational theories and can promote learning. The chapter Algorithms dealt with the task of creating a set of variables for a desired outcome. This chapter deals with the ways in which these variables are dependent on one another, and explores the effects they have on each other and on the systems as a whole.

Visual parametric structures are complex systems, in which the connections between variables, defined by a network of conditional statements, are as influential as the variables themselves in determining the final outcome. Writing the parameters defines the exploration field while manipulating them allows for infinite variations within it. It is a highly dependent system, in which the complex connections between variables determine its essence, and each variable has the power to influence the system as a whole. Changing one detail can cause a chain reaction, changing the system entirely. In the code 'Embedded Iteration' for example (Fig.1), all the images where created by the same code. Changing just one line in the code is what creates their significant differences. (Reas 2010, 64). The adjustable quality of parametrically designed systems offers participation, which as demonstrated in the previous chapters has a positive effect on learning. In addition, the dependency between the system's variables acts as a fertile ground for exploring cause and effect.

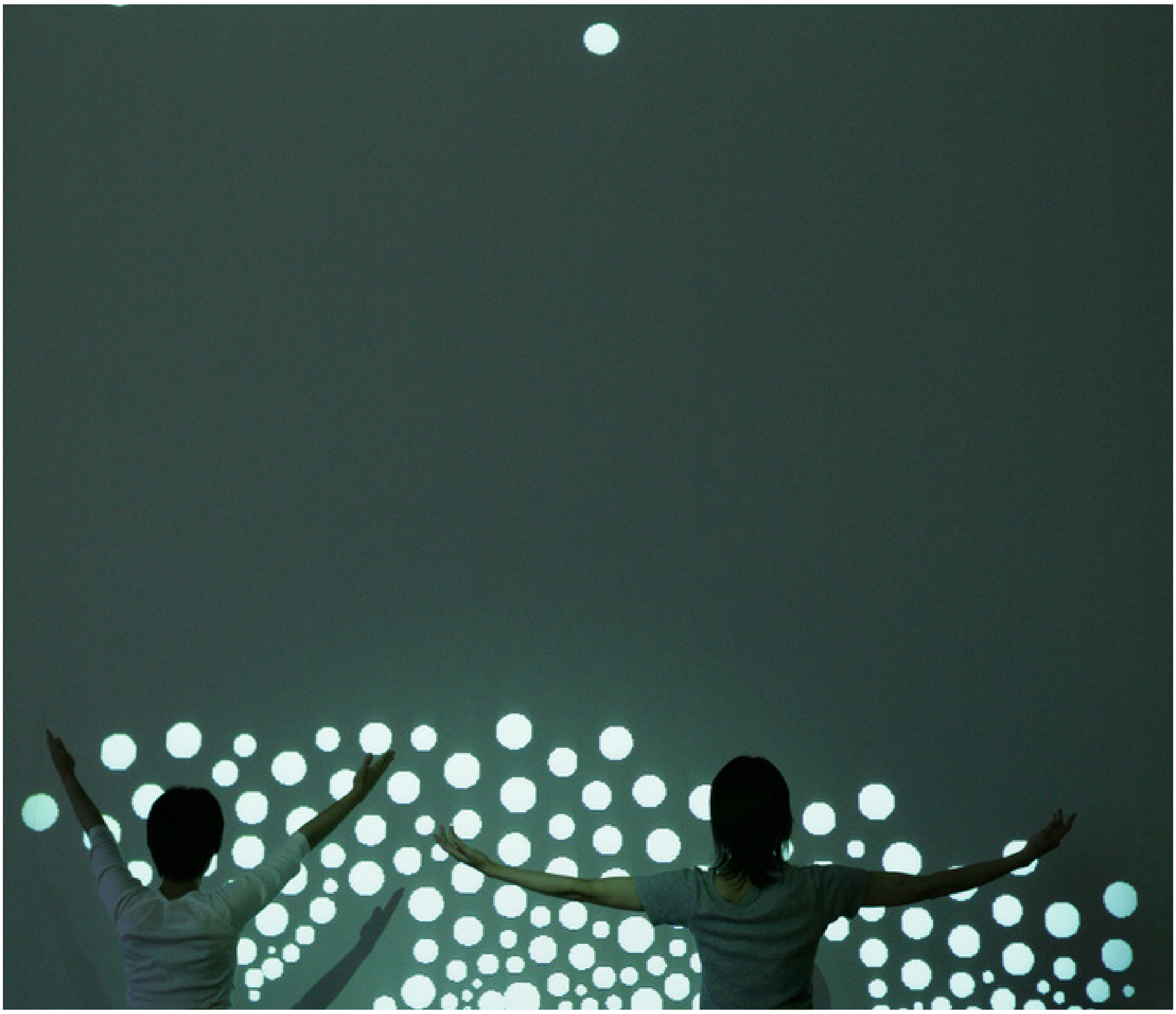
Dependency as a parametric theme has the potential of facilitating and enhancing learning on two different levels. First, similarly to other discussed themes, it allows personal participation. Designing using a parametric platform enables navigating through and manipulating connections between variables, in order to explore endless options, and to influence the process personally. It thus allows a personalization of the engagement experience, and relates to the notion that the best learning occurs when the learner takes charge of the learning process. (Papert 1993, 25). However, dependency offers an additional, unique learning opportunity, rooted in its essence. The ability to affect the whole system, by engaging with only one of its aspects at a time, crystalizes the significance each individual variable holds. It facilitates learning about cause and affect, which are at the root of any system. It provides an understanding of the multiple options for connections and relationships between variables. Connecting new data with existing knowledge is instrumental in rooting, securing and fastening information. Additionally, the fact that the connections and relationships between variables are visible and are open for manipulation and change, provides an opportunity of creating a personalised set of connections and conditions. Seymour Papert writes about the importance that connection making has to a learning process. He discusses his personal experience learning about flowers, and emphasises the importance of creating personal connections between details of information. He discusses the connections as essential and instrumental, acting almost as glue in constructing meaning. Papert claims that the ability to associate different ideas to each other is the ability to fully understand them. He writes: "…The new connections supported one another more effectively and they were more and more long lasting… Indeed, the description "connectionism" fits my story better than "constructivism". It even suggests that the deliberate part of learning consists of making connections between mental entities… This suggests a strategy to facilitate learning by improving the connectivity in the learning environment." (Papert 1993, 104-5). Parametric systems are dependent and are open for personal reconstruction of their inner connections. Their dependent nature allows personal construction of complex conditional networks. Connecting new information with existing knowledge, the learning process is enriched and enhanced. It becomes a process of personalising a vast network of connections and meaning, strongly rooting information with each connected anchor.

07.1 Funky Forest (2007) Design I/O

Funky Forest in an ecosystem where participants are able to manage resources in order to influence the environment around them. For example, streams of water flowing across the floor can be diverted to encourage different parts of the forest to grow. When a tree does not receive enough water, it will begin to wither away. By pressing one's body into the forest, new trees are created based on the manner of pressing, body-shape and behavior. As visitors explore and experiment they discover that their environment is inhabited by life forms that depend on a thriving ecosystem to survive. The health of the trees contributes to the overall health of the forest and the types of creatures that inhabit it. (Reas 2010, 172).

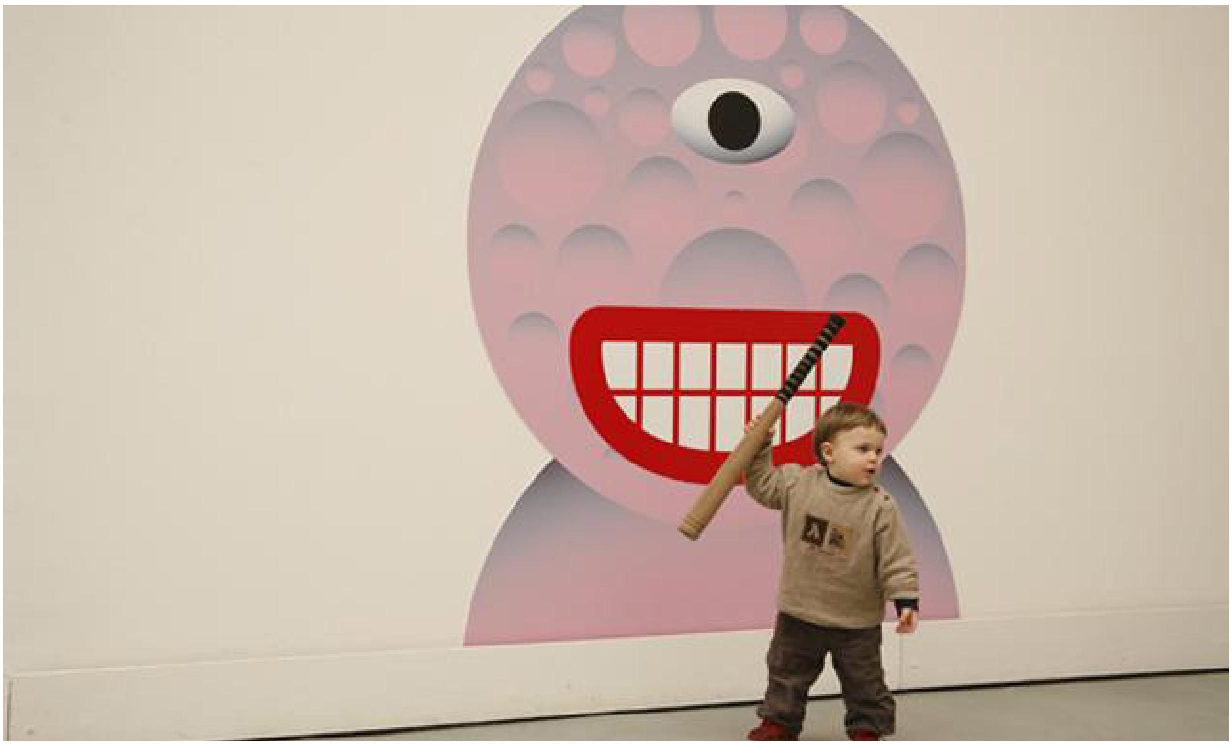
Funky Forest is a reactive system. Influencing every part produces a reaction, changing the balance of the system. If a tree does not receive enough water, it will wither away. The unhealthier trees are, the less birds in the forest skies. Fewer birds mean too many insects that can harm the plants, and so on. *Funky Forest* can teach children about the rules of nature. But more than that, is can teach everyone about relative relations, about cause and effect, about the role of an individual within a system, and about dependable networks. If we examine parametric design platforms as systems of information, then the connections between its parts is what defines its specific meaning at any given moment. Funky Forest is such a system, in which the network of information is open for reconstruction. It facilitates and encourages the formation of personal information structures. Creating personalised information systems can promote better understanding. (Papert 1993, 103). Linking and associating segments of information with existing knowledge supports their meaning and enforces their existence. Repetition also provides an opportunity for observing the system both as a whole and as a compilation of individual segments. It allows for an experience that Edith Ackermann calls "both stepping in and stepping out" and claims that both are equally important in reaching deeper understanding. (Resnick et al. 1996, 29).

07.2 Footfalls (2006) Golan Levin & Zachary Lieberman



In Footfalls, stepping and stomping sounds produced by the visitors' feet are detected by microphones under the floor, and used to govern the size and number of virtual objects that fall from a six-meter high projection. The harder the visitors stomp, the more items fall. Using their silhouettes, visitors can then "catch" and "throw" these projected objects around. (Reas 2010, 99). The dependent system of *Footfalls* provides an experimental experience, examining the networks that construct the installation. The action of stepping influences the falling items, stepping and waving relate to each other, the movement of the items affect the behaviour and position of the items that come next. It is a reliant and connected system that allows investigation by providing an opportunity to examine its connections and their meaning. Similarly to the previous example, there is more than one manner of investigation. One can explore the behaviour of one segment of the system, its influence on the system as a whole and the ways in which it is affected by movement and stepping sounds, and by the other particles. Additionally, the system allows a broader exploration angle, through examination of the system as a whole. It provides an option of shifting between view angles, deepening learning and understanding. (Resnick et al. 1996, 29). This type of information connections do not only promote better learning of the information, but also allows for a wider understanding, that spreads to many directions. (Mindstorms). Thanks to the reliant nature of the connection system, we can understand how the balls fall, how they relate to each other, and how our behaviour influences it all.

07.3 Pianeti Sonori (2006) Luca De Rosso & Alberto Moro (with Studio Camuffo)



Pianeti Sonori is an installation of several giant reactive monsters. Buttons, installed behind each monster, are connected to a computer programme made using the language Processing. According to how children engage with the monsters, the software changes their expression and sound. (Reas 2010, 123).

The monsters can smile or frown, sing happily or make upset sounds, dependent on the rhythm, manner and the intensity of the pressure inflicted on the buttons.

Its an example of how the chain reaction feature embedded in the code, is translated into the experience. When a monster is being treated aggressively it changes its behaviour into a threatening one. This will in turn evoke an angry reaction from the child and so on. The immediate and direct nature of these reactions in an example for the ways in which connectivity has the potential to promotes learning on several levels. The way in which a monster reacts to a specific gesture can provide information about the ways in which others might react to it. It is also a window of observation into the relationship between gestures and reactions on a more general level, revealing the cause and affect of behaviour. The users are invited to explore the relationships between the behaviour of the monsters and their own, by observing the changes inflicted on the monsters by manipulating the system.

The quality of connectedness and of continuity has been demonstrated as powerfully conductive to learning. (Papert 1993, 145). The three examples discussed in this chapter demonstrate such continuous systems, in which every detail of the information exists in relation to, and with direct connection with, all the other details that form the whole. Information is not expected to be consumed in a vacuum but rather as a part of a transparently dependent system, emphasising the connections between particles rather then ignoring them.

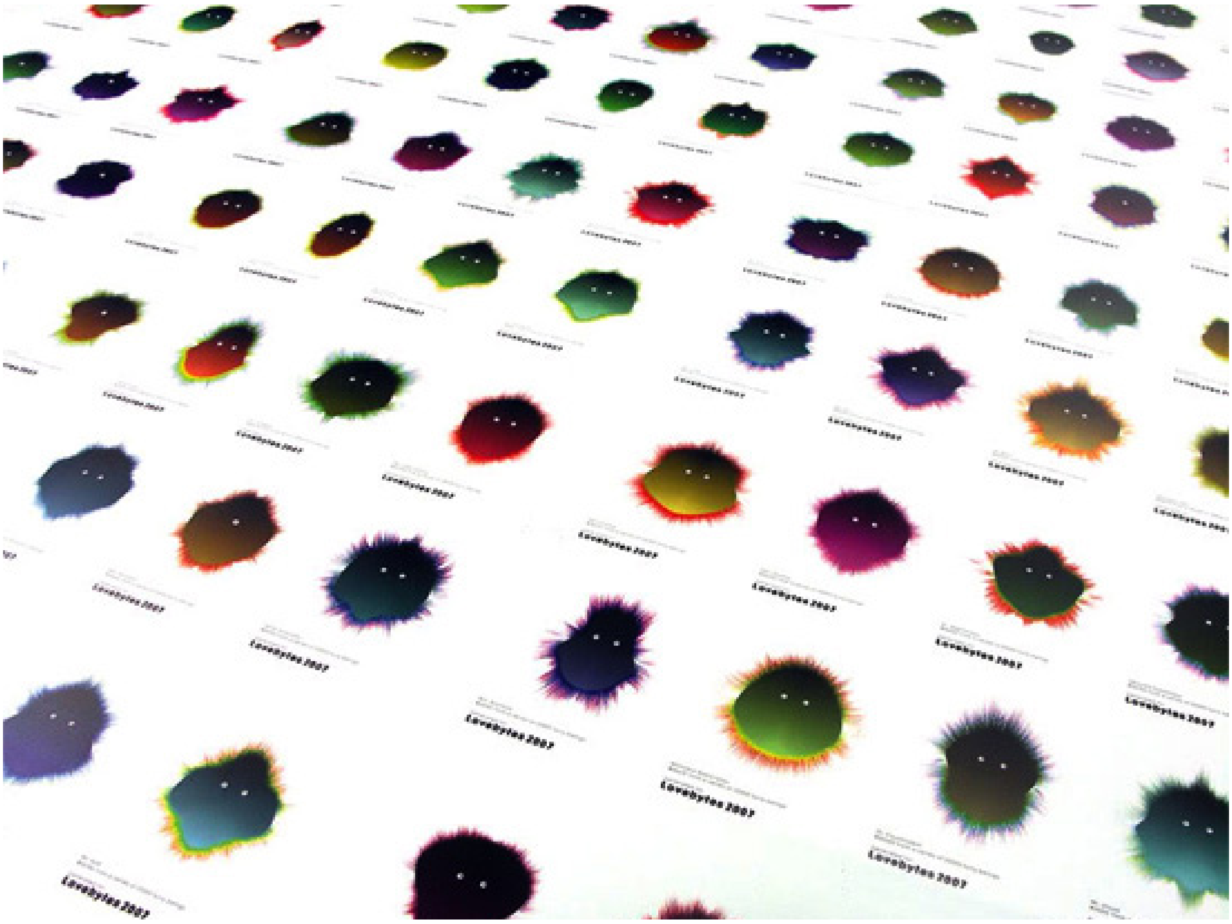
Chapter 08
Random

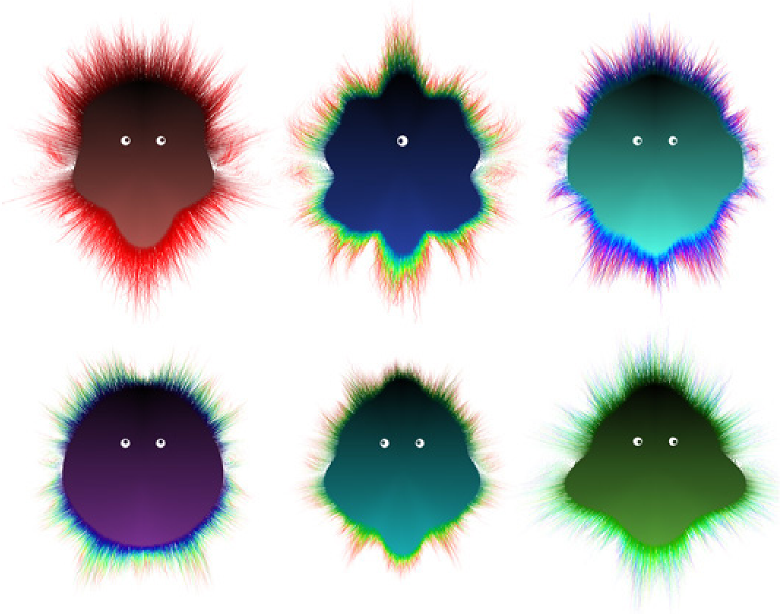
This chapter deals with the use of random generation of variables in parametric design systems. It explores the manifestation and uses of random code choices, and its connection to ideas of learning through experiment, tinkering and play.

The command Random is very common in programming. It enables the generation of an unplanned and irregular range of possible outcomes within certain constraints. Randomness in code will be defined against a set of limitations, defining a starting point and an end point for the command random to be applied. The code can act randomly between certain variables; the differences in their values will determine the range of activity. The variables can be anything from a number to a shape to a location point and so on. It can be applied to one aspect of a system or to all of its segments. There is a range of possibilities and degrees to which it can be used, for a variety of purposes. Randomness is a useful tool in exploring unexpected compositions and solutions. It is often used for generating numerous amounts of options, in order to be able to choose an optimised solution. It can also be used to emulate unpredictable qualities of our physical reality. For example, sequences of random numbers that differ only slightly from one another is a technique that is often used for simulating natural effects like wind, waves and rock formations. (Reas 2010, 103). Exploring possibilities with the command random is an experience in which the end result is minor to the process. The results are randomly generated and hence unknown and unpredictable. Creating with randomness highlights explorations rather than solutions. It supports a playful, experimental approach. The word tinkering is often used in reference to a playful manner of engagement. Mitchel Resnik defines tinkering as "...An approach characterized by a playful, exploratory, iterative style of engaging with a problem or project. When people are tinkering, they are constantly trying out ideas, making adjustments and refinements, then experimenting with new possibilities, over and over". (Honey et al. 2013, 164). Resnik discusses tinkering as similar to Bricolage, in that it involves spontaneous improvisation with available materials rather than a detailed execution of a plan. Similarly to bricolage, tinkering echoes constructivists theories about the importance of experimentation and engagement to learning. (Honey et al. 2013, 165). Piaget claimed that play is the work of children, in that it is a process of intellectual growth. (Resnick 2007). Papert writes about utilizing a playful approach to the solution of problems as extremely valuable. He claims that thinking about the problem, rather than exercising rules to solve it, is what fosters learning. He writes: "Any kind of 'playing with problems' will enhance the abilities that lie behind their solution". (Papert 1993, 87).

The following examples will illustrate the ways in which designing parametrically with random variables is a valuable tool for learning. Being a digital tinkering process, it prioritizes creativity and playfulness over planning and optimization, enabling an intuitive learning process.

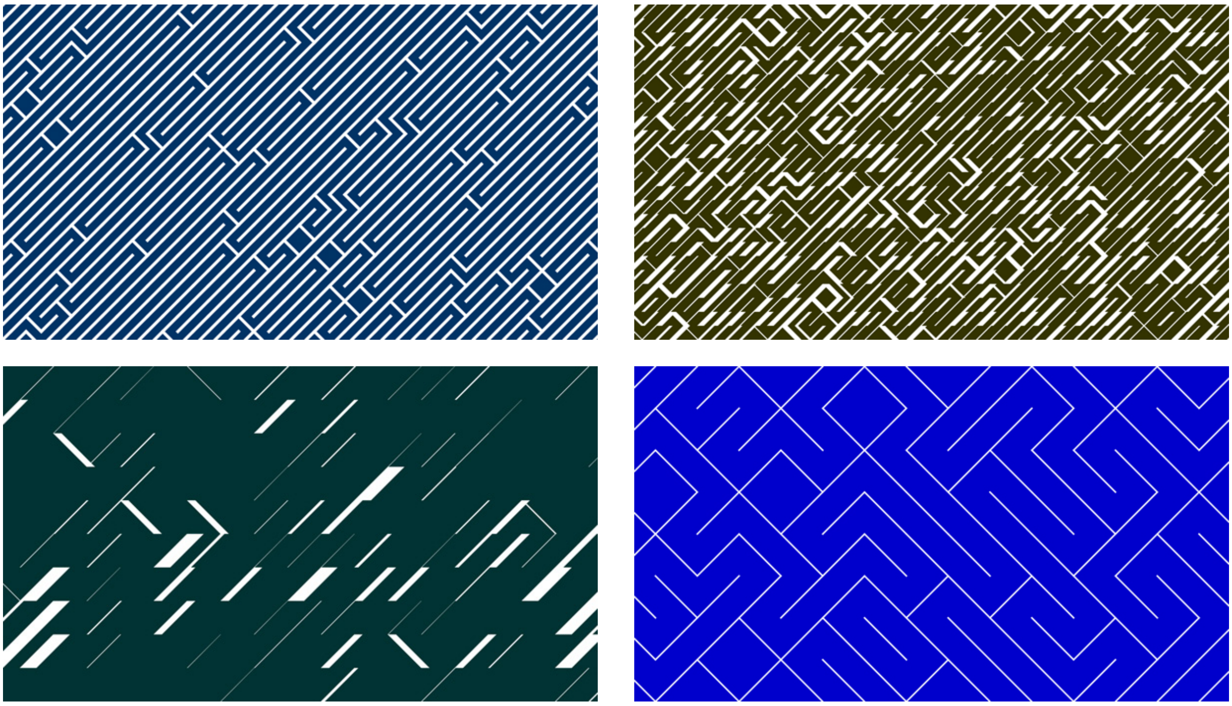
08.1 LoveBytes (2007) Universal Everything





Lovebytes Is a series of 20,000 characters, all constructed from the same basic elements. The project is described in the design firm's website: "Exploring the limitless permutations of a single-cell organism, *Lovebytes* creates 20,000 unique variations on a basic amoeba character with the aid of generative software." (www.universaleverything.com, 30.01.2014). All *Lovebytes* share some features: they all have the same eyes and the same type of hair. The randomness that was written into the computer code generating the *Lovebytes*, has created a notable amount of variation between them. *Lovebytes* are a family of 20,000 individuals, similar yet very different. The values that are chosen randomly by the software are the colour of their hair and its length, the size and shape of their bodies and their number of eyes. These variables have managed to generate a remarkable amount of characterful, individual *Lovebytes*. The random feature enables a playful exploration of all the possible combinations and variations achievable. It is a process that centralises experimentation and fun. Simultaneously, it offers the results of a profound research by its ability to explore vast amounts of combinations and suggestions. The spontaneous and unpredictable nature of the search frees from constraints of a final outcome and might enable innovative solutions. Carol Strohecker tells about her experience in her Knot Laboratory at MIT, and the benefits of tinkering with knots: "Thinking spanned the deliberate and spontaneous, the rational and affective, the conscious and unconscious" (Turkle 2007, 25). Random actions in code such as *Lovebytes* are of a similar nature. It might start with a certain deliberate aim, such as generating a family of colourful creatures. The process will then be randomized, turning unpredictable and unintentional. Once a result is generated, it can be examined and intentional decisions about the variables can be made. The process will then enter another random cycle and so forth: It allows combining wilful and accidental approaches. Playful and experimental, the random process navigates through a range of variables as well as through a range of ways of thinking.

08.2 Yes No (Software 1) (2012) Casey Reas



Yes No is a software that is constantly generating random patterns. Following a set of guidelines, the program chooses several values randomly, resulting in different outcomes. The program is described in the designer's website: "It is an exploration into randomness, but focused through narrow constraints. For each unit or grid, the software makes a decision to draw one kind of line or another. The software constructs a new composition every other second." (www.reas.com, 30.01.2014). The background colour accompanying the lines is also selected randomly. The compositions created by *Yes No* are extremely varied; the random range was defined widely. Watching the program producing the colourful compositions is a playful experience. They appear on the screen, surprizing and unexpected. Random processes are explorative platforms. Yes No is a research into the possibilities of lines and it might evoke thoughts about length and width, continuity and separation, density and sparsity, complexity and simplicity. These thoughts might evoke additional experiments with the software. In Yes No the research activity is the final outcome. The compositions, temporary and altering, are only a mean to end and not an aim on their own. It is an open experiment, with no specific goal: It is a playful, enjoyable engagement opportunity. As illustrated in the previous example, it is a process that is as investigative as it is fun, further demonstrating the link between playing and learning. Mitchel Resnik writes about the connection between the two: "Play and learning can and should be intimately linked. Each, at its best, involves a process of experimentation, exploration, and testing the boundaries".

(Resnick 2007, 3).

08.3 Love Blossoms (2011) Daniel Brown

Love Blossoms is an online platform exhibiting parametrically designed flowers. Participants are invited to choose a seed from a pile and watch a unique flower come to life. The flowers can then be emailed to friends and family. The program randomises values such as speed and direction of growth, number of stems and of flowers, and petal texture. The project is described in Dezeen magazine: "Love Blossoms generates digital flowers, using generative programming in order to make it come to life and appear to grow in front of the viewer. The design is such that each flower grows uniquely, 'choosing' its own organic shape as it would in nature. Each flower is generated so that no two blooms are ever the same." (www.dezeen.com, 30.01.14). The result is a platform that creates numerous variations, which do not only differ in the way they look, but also in the manner and rhythm in which they grow. The randomly chosen variables generate unique flowers, which vary from one another in the way they look and behave. They are digitally growing from a seed to a flower, in a playful and unexpected manner, while the generating code offers opportunity for further exploration. Altering the random range will produce more irregular results, while adding constant or random variables is a way of exploring the platform even further. Engaging with a random visual system has the potential of being a fertile playground. By combining randomised aspects with active participation of choosing and manipulating the parameters, it is an idealised learning platform. The flowers might grow too slowly, or not vary enough in colour. They might be too large or too small, too similar or too far apart. The ability to test out ideas, while tinkering with the subject, allows for a calm investigative experience. Papert writes about a playful problem solving process: "...Spending relaxed time with a problem leads to getting to know it, and through this, to improving one's ability to deal with other problems like it". (Papert 1993, 87).

Random generation of code offers generation of endless playful possibilities on any given set of guidelines. It can be used to produce joyful experiences and range of solutions. Explorative approach makes it an investigative process, which is directly related to learning.

Chapter 09.
Utilization

Throughout this thesis I have demonstrated the various ways in which the practice of parametric design and the act of learning intertwine. By exploring inherent parametric design themes in relation to constructivists learning theories, I have claimed that the constraints and possibilities presented while designing parametrically create a fertile ground for acquiring skills and information.

I believe that the options for translating these concepts into visual learning platforms are endless. The profound learning potential designing parametrically holds, spreads across disciplines and audiences. I believe that through experiencing the engaging and participatory interaction of designing parametrically, any subject can be learned more successfully. Dealing with shape and compositions, biology and chemistry, philosophy or any other subject, the inclusion of thinking with algorithms, simulations, repetitions, reactions and random greatly enhance and improve the learning process.

The experience provided by engaging with parametric design systems strongly relate to constructivist learning theories in many ways, which have been demonstrated throughout this thesis. These systems provide an opportunity of transformation abstract notions into concrete representations. By transforming ideas into existing objects open to manipulation it allows a profound exploration and experiencing of an idea. These platforms also offer participation in the learning process, allowing personal exploration and manipulation of information. Finally, they provide a playful and experimental experience. Centralizing the process rather then the final outcome, these platforms are an investigative platform leaving room for error and surprise.

In conclusion, I believe that parametric design platforms can be used to enhance the learning process of any subject. They allow dealing with massive amounts of information or with just two variables: Whatever the subject or tools, these platforms provide a stepping stone for deeper understanding.

From general skills such as differentiating between essential and secondary information, to more specific knowledge such as acquiring a new language or understanding a complex concept or system, the possibilities are endless. Providing a flexible and affluent experience, parametric design platforms offer a truly exciting learning opportunity.

Bibliography

Books and Articles:

Adler, Richard P., and Seely Brown, John. 2008. “Minds on Fire: Open Education, the Long Tail, and Learning 0.2”. EDUCAUSE Review, vol. ٤٣, no. 1: 16–32

Baars, Bernard J., Ramsoy, Thomas Z., and Laureys, Steven. 2003 “Brain, Conscious Experience and the Observing Self”. *Trends in Neurosciences* Vol. 21 No. 12: 671-675

Bohnacker, Hartmut., Gross, Benedikt., Laub, Julia., Lazzeroni, Claudius. (Editor) 2012. *Generative Design: Visualize, Program, and Create with Processing.* New York: Princeton Architectural Press.

Brown, Stuart., and Vaughan Christopher. 2009. *Play: How it Shapes the Brain, Opens the Imagination, and Invigorates the Soul*. New York: Penguin Group

Brown, Stuart L. (Interview). 2009.“Discovering the Importance of Play through Personal Histories and Brain Images”. American Journal of Play Vol. 4. (2009): 399-412

Dougherty, Dale. 2009. “Maker Friendly.” MAKE, vol. 03,:7

Dweck, Carol. 2000. Self-theories: *Their Role in Motivation, Personality, and Development (Essays in Social Psychology).* United Kingdom: Psychology Press

Dewey, John. 1938. *Experience and Education*. USA: Kappa Delta Pi

Dewey, John. 19xx. *How We Think*. USA: Kappa Delta Pi******

Fischer, Gerhard. 2011. “Understanding, Fostering, and Supporting Cultures of Participation.” *Interactions*. Vol. XVIII.3: 42-53

Fry, Ben., and Reas Casey. 2007. *Processing: A Programming Handbook for Visual Designers and Artists*. Massachusetts: MIT Press

Fry, Ben. 2008 *Visualizing Data: Exploring and Explaining Data with the Processing Environment*. Canada: O'Reilly Media, Inc.

Gibson, James J. 1986. *The Ecological Approach to Visual Perception*. New Jersey: Lawrence Erlbaum Associates, Inc.

Honey, Margaret, and Kanter, David E. 2013. (editors), *Design, Make, Play: Growing the Next Generation of STEM Innovators*. New York: Routledge

Klanten, Robert. 2011. *A Touch of Code: Interactive Installations and Experiences*. Edited by Sven Ehmann and Verena Hanschke. Location: Germany: Die Gestalten Verlag

Kolodner, Janet L., Camp, Paul J., Crismond, David., Fasse, Barbara., Gray, Jackie., Holbrook, Jennifer., Ryan, Mike., and Puntambekar, Sadhana. "Problem-Based Learning Meets Case-Based Reasoning in the Middle-Scholl Science Classroom: Putting Learning by Design Into Practice". The Journal of the Learning Sciences. 12 (2003): 495-547

Maeda, John. 2000. *Maeda @ Media*. New York: Rizzoli

Maeda, John. 2001. *Design by Numbers*. Massachusetts: MIT Press

Maeda, John. 2004. *Creative Code: Aesthetics + Computation*. London: Thames & Hudson

Malafouris, Lambros. 2013 *How Things Shape the Mind, A theory of Material Engagement*. Massachusetts: MIT Press

Moggridge, Bill. 2007. *Designing Interactions*. Massachusetts: MIT Press

Millner, Amon. 2012. "Computer as Chalk: Supporting Youth as Designers of Tangible User Interfaces." *Constructionism 2012: Theory, Practice and Impact*: 339-349

Mizuko, Ito. et al. 2013. *Connected Learning: An Agenda for Research and Design*. Irvine CA: Digital Media and Learning Research Hub.

Norman, Don. 2013 *The Design of Everyday Things (Revised and Expanded Edition)*. Philadelphia: Basic Books

Norman, Kent L. 2008. *Cyberpsychology: An Introduction to Human-Computer Interaction*. New York: Cambridge University Press

Piaget, Jean. 1973. *To Understand is to Invent*. New York: Grossman Publishers

Papert, Seymour. 1993. *The Children's Machine: Rethinking School in the Age of the Computer*. New York: BasicBooks

Papert, Seymour. 1993. *Mindstorms: Children, Computers, and Powerful Ideas*. 2nd Edition. New York: Basic Books

Patrik Schumacher in conversation with Georgina Day. "On Parametricism" *Paper – Platform for Architectural Projects, Essays & Research* 4 (2012)

Pessoa, Luiz., 2013. *The Cognitive-Emotional Brain: From Interactions to Intergration*. Massachusetts: MIT Press

Reas, Casey., and McWilliams, Chandler. 2010. *Form + Code in Design, Art, and Architecture*. New York: Princeton Architectural Press

Resnick, Mitchel, and Silverman, Brian. 2005. **Some reflections on designing construction kits for kids**. *Proceedings of the 2005 conference on Interaction design and children*. New York: ACM

Resnick, Mitchel., Kafai, Yasmin B. (Editors). 1996. *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. New Jersey: Lawrence Erlbaum Associates, Inc.

Resnick, Mitchel. "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten" *Creativity & Cognition Conference* 1 (2007)

Resnick, Mitchel. 2002. "Rethinking Learning in the Digital Age." *The Global Information Technology Report: Readiness for the Networked World*. edited by G. Kirkman. New York: Oxford University Press.

Resnick, M., and Silverman, B. "Some Reflections on Designing Construction Kits for Kids". *Proceedings of Interaction Design and Children conference*, Boulder, CO. (2005)

Resnick, M. "Lifelong Kindergarten". *Cultures of Creativity*. LEGO Foundation (2013)

Resnick, M. (2012). "Reviving Papert's Dream". *Educational Technology*. vol.52 (2012): 42-6

Schon, Donald A. 1983. The Reflective Practitioner. USA: Basic Books

Shiffman, Daniel. 2012. *The Nature of Code: Simulating Natural Systems with Processing*. Self-Publication.

Singer, D., Golikoff, R., and Hirsh-Pasek, K. (Editors). 2006. *Play = Learning. How Play motivates and enhances children's cognitive and social-emotional growth*. New York: Oxford University Press.

The Design-Based Research Collective. "Design-Based Research: An Emerging Paradigm for Educational Inquiry". Educational Researcher 32 (2003): 5-8

Turkle, Sherry, and Papert, Seymour. 1990. "Epistemological Pluralism and the Revaluation of the Concrete." Signs, vol. 16, no. 1

Turkle, Sherry. (Editor). 2008. *The Inner History of Devices*. Massachusetts: MIT Press

Turkle, Sherry. (Editor). 2007. *Evocative Objects: Things We Think With*. Massachusetts: MIT Press

Turkle, Sherry. 1984. *The Second Self: Computers and the Human Spirit*. New York: Simon & Schuster, Inc.

Turkle, Sherry. (Editor). 2008. *Falling for Science: Objects in Mind*. Massachusetts: MIT Press

Woodbury, Robert. 2010. Elements of Parametric Design. Oxon: Routledge

Websites:

http://makezine.com/ (January 2014)

http://www.jonasheuer.de/ (February 2014)

http://arandalasch.tumblr.com/page/5 (December 2013)

http://roberthodgin.com/magnetosphere/ (January 2014)

http://roberthodgin.com/bait-ball/ (January 2014)

http://www.legildesign.com/archives/680 (January 2014)

http://formandcode.com/ (November 2013)

http://www.generatorx.no/ (December 2013)

http://www.universaleverything.com/works/ (January 2014)

http://www.fastcodesign.com/ (January 2014)

http://www.yeoahn.com/ (January 2014)

http://www.danielbrowns.com/ (December 2013)

http://www.play-create.com/ (January 2014)

http://www.flong.com/ (January 2014)

http://www.universaleverything.com/ (January 2014)

http://vacuum-mag.tumblr.com/ (January 2014)

http://everyware.kr/ (December 2013)

http://josephlgriffiths.com/ (January 2014)

http://design-io.com/ (January 2014)

http://www.theowatson.com/ (January 2014)

http://www.visualcomplexity.com/ (November 2013)

http://infosthetics.com/ (January 2014)

http://www.scientificamerican.com/ (December 2013)

http://processing.org/ (October 2013)

http://www.dezeen.com/ (January 2014)

http://www.itsnicethat.com/ (January 2014)

http://www.reas.com/ (December 2013)

http://fernandoorellana.com (January 2014) http://nevolution.typepad.com/ (January 2014)

http://www.edutopia.org/ (February 2014)

http://dkroy.media.mit.edu/ (November 2013)

http://www.makered.org/ (January 2014)