

LumiCAD

*Interactive Organic
Generative Design for
Personal Photonics
with OLEDs*

Bachelor's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University



by
Salsabil Hamadache

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr.-Ing. Kowalewski

Registration date: 26.06.2015
Submission date: 29.07.2015

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, July 2015
Salsabil Hamadache

Contents

| | |
|-----------------------------------|----------|
| Abstract | xv |
| Überblick | xvii |
| Acknowledgements | xix |
| Conventions | xxi |
| 1 Introduction | 1 |
| 2 Related work | 3 |
| 2.1 Generative Design | 3 |
| 2.2 Inspiration | 5 |
| 2.2.1 Nervous System | 9 |
| 2.2.2 Patrick Jouin | 12 |
| 2.3 L-Systems in Nature | 13 |
| 2.4 The Superformula | 14 |
| 2.5 OLEDs | 16 |

| | | |
|----------|---|-----------|
| 2.6 | Processing | 16 |
| 2.7 | Summary of Related Work | 17 |
| 3 | Own work | 19 |
| 3.1 | The growing process | 20 |
| 3.1.1 | Simple straws | 20 |
| 3.1.2 | Creating Wreaths | 23 |
| 3.1.3 | Getting More Complex | 24 |
| 3.2 | The Flower | 25 |
| 3.2.1 | Import Problems | 25 |
| 3.2.2 | Non-Basic 3D Shapes in Processing | 26 |
| 3.3 | The Process of Printing the Model | 31 |
| 3.3.1 | Leaves | 31 |
| 4 | Evaluation | 33 |
| 4.1 | System Usability Scale | 33 |
| 4.2 | User Study: Part I | 34 |
| 4.2.1 | Results of the background-based questions | 34 |
| 4.2.2 | Results of the questions taken from the SUS | 34 |
| 5 | Summary | 37 |
| 5.1 | Summary | 37 |
| 5.1.1 | One final result | 37 |

| | | |
|----------|--|-----------|
| 5.1.2 | Lessons learned | 40 |
| 5.1.3 | Some Things to Note | 40 |
| 5.2 | Future Work | 41 |
| A | Evaluation Details | 45 |
| A.1 | Exact format and results of the SUS user study | 45 |
| | Bibliography | 51 |
| | Index | 53 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Work in progress: screen shots of the running program | 2 |
| 1.2 | Work in progress: screen shots of the finished models | 2 |
| 2.1 | top-left:Dah, below: Vas ; right: 610 | 7 |
| 2.2 | more 3D-printed lamps by Kytäinen mor | 7 |
| 2.3 | left:real Dahlia, right: real coneflower(610) | 8 |
| 2.4 | optimal spiraling with golden ratio | 9 |
| 2.5 | generatively designed lamps by Nervous System | 10 |
| 2.6 | The design process for the 'reaction lamp' (first one in Fig. 2.5) ; Nervous System | 10 |
| 2.7 | 'Laplacian Zoetrope' and 'Hyphae Zoetrope' ; Nervous System | 11 |
| 2.8 | 'hyphae 3D 2' and 'hyphae crispata 1' ; Ner | 11 |
| 2.9 | 'Bloom' ; Jouin [2010] | 12 |
| 2.10 | Exemplary designs for a 3D Printed Lamp Design Challenge Kurtsev [2010] | 12 |

| | | |
|------|--|----|
| 2.11 | 2 models of trees competing for light ; Prusinkiewicz [2004] | 14 |
| 2.12 | 3D models of flowery shapes created by the superformula | 15 |
| 3.1 | Paper sketches of the first idea | 19 |
| 3.2 | Growing straws in the first phases of the program | 22 |
| 3.3 | Combining straws to a bouquet | 23 |
| 3.4 | Bug creating a wreath | 23 |
| 3.5 | Superformula creating a supershape resembling a flower | 28 |
| 3.6 | Superformula creating another supershape resembling a flower | 28 |
| 3.7 | "Drawing" a sunflower | 29 |
| 3.8 | Finished model of a lamp ready to 3D print . | 30 |
| 3.9 | 3D model from different perspectives | 30 |
| 3.10 | another 3D model from different perspectives | 31 |
| 3.11 | Screen shots taken of the designing environment before exporting the models from above | 31 |
| 3.12 | Before vs after: one leaf and plants with the respective leaves | 32 |
| 5.1 | Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs | 37 |
| 5.2 | Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs | 38 |

- 5.3 Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs 39

List of Tables

| | | |
|-----|--|----|
| 4.1 | System Usability Scale results | 35 |
|-----|--|----|

Abstract

Adding interactivity to 3D printing and creating organic designs with code is the main goal pursued with this thesis. With generative design, the Processing programming language, the studies of natural shapes and patterns, and personal fabrication all mingled together, this goal seemed to be an interesting one and exceeded all expectations. Writing a program in which an arbitrary user can create his own plant-shaped object and 3D print the result required research mainly in the field of trigonometry and computer graphics in general, but also research in nature and natural processes. The program makes virtual insects grow a plant by walking in algorithmically defined, yet randomized ways and leaving matter on their way, which the plant will consist of. In playful ways, the user can interact with the growing by creating bugs, killing them or by destroying their traces. He can add leaves and petals to the plant. This results in an individualized and aesthetic lamp illuminated by OLED tiles that produce organic light for the organically shaped plant.

Überblick

Künstlern und Bastlern wurden durch die Erfindung von 3D-Druckern und Lasercuttern Türen zu Projekten geöffnet, die in Handarbeit viel Zeit und Ressourcen kosten würden. Dafür können diese jedoch den Umgang mit Computern nicht vermeiden. Um einen Gegenstand zu designen und in drei Dimensionen auszu-drucken, muss am Computer eine sogenannte STL Datei erstellt werden, beispielweise mit FreeCAD oder Blender. Dann wird diese an den Drucker geschickt und ausgedruckt. Am Druckprozess selbst ist der Künstler jedoch nicht beteiligt, weswegen man 3D Drucken nicht als interaktives Designen bezeichnen kann. Es wäre jedoch ein Schritt zur Interaktivität hin, wenn man zumindest auf dem Computerbildschirm Interaktion anbietet. So könnte ein Tool, mit welchem auch ein Laie selber ein wenig rumspielen kann, auch wenn er von Design, FreeCAD oder Computern generell, nicht viel Ahnung hat, Gegenstände entwerfen kann. Mit diesem Ziel wird in dieser Arbeit ein Programm geschrieben, das die Möglichkeit bietet, organische Lampen zu kreieren. Hierbei wächst Materie zunächst von selbst, kann aber durch den Nutzer gelenkt, gefördert und gebremst werden. So entsteht bei jedem Durchlauf ein einzigartiges Muster, in dem generative Algorithmen leiten und ihre Parameter durch den Nutzer bestimmt werden. Die Lampe, die gedruckt wird, ist also Kunst, die geteilt durch den Künstler und durch den Computer entsteht.

Acknowledgements

Firstly I would like to thank my supervisor Dipl.-Inform. René Bohne for his ideas and valuable guidance. Of course I also want to thank Prof. Dr. Jan Borchers for the opportunity to write this thesis at his chair and him and Prof. Dr.-Ing. Kowalewski for examining it.

Thanks to the whole chair for supporting me in writing this bachelor thesis in many different ways. I am especially grateful for the help of the FabLab staff, in particular Jan Thar for his help with printing the final product. Furthermore, I would also like to say my thanks to fellow student Christian Frohn for his endless readiness to help out.

At last, I would like to thank my family for their support throughout my studies.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:

Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in Canadian English.

Download links are set off in coloured boxes.

File: [myFile](#)^a

^ahttp://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

In this bachelor thesis a program will be introduced in which even an arbitrary non-expert user with no prior experience can “grow” an individual aesthetic illuminated plant, which can be 3D printed and turned into an actual lamp in the context of personal fabrication, smart lighting and generative design.

In an interactive and virtual environment the user watches virtual insects create a plant by walking around in the three-dimensional scene in algorithmic manners and carrying the matter to build up the plants with behind them. He can interfere by creating and killing the bugs and adding blooms and leaves and destroy paths in hindsight. Unique but regular patterns are the outcome of the creation process that happens in the Processing programming environment. This elaboration explains the motivation, gives a brief overview about the work field and the status-quo of it and then guides through the implementation process. All mathematical and other backgrounds are given insights into throughout the thesis. Finally, the program is evaluated and an outlook on possible extensions is given.

Here, screen shots of the running program are depicted in Figure 1.1 and in Figure 1.2 one can see resulted models ready to print.

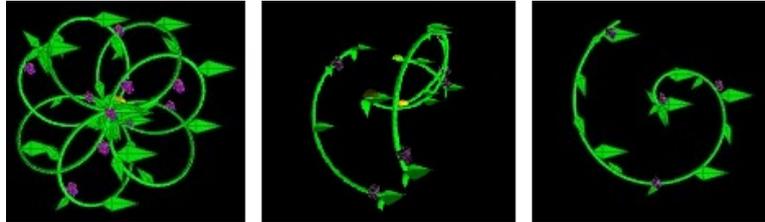


Figure 1.1: Work in progress: screen shots of the running program

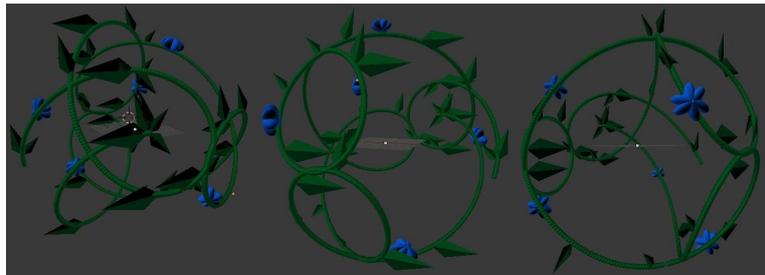


Figure 1.2: Work in progress: screen shots of the finished models

Chapter 2

Related work

In this chapter a brief overview about the work of others in the field this thesis ranges in is given and both theoretical backgrounds retrieved so far and designs already created will be presented.

2.1 Generative Design

At the outset, it is important to understand the main theme behind this work, which is Generative Design.

Jon McCormack et al. [2004] (with Alan Dorin and Troy Innocent) at the Monash University in Australia give an overview about Generative Design in their paper. It gives a helpful insight into what Generative Design is, it was used to get a clear definition to work with in my thesis, and it gives numerous answers to the question how Generative Design can look like. The authors suggest that there can be different processes to create designs. Their definition underlines the following aspects:

What is Generative Design?

Generative systems offer a methodology and philosophy that view the world in terms of dynamic processes and their outcomes. [...] Conceptualisation shifts from the primacy of objects to envisaging interacting components, systems

and processes, which in turn generate new artefacts.

They come up with a connection to synthesis as in natural systems and state that something is to be called generative if it generates novelty and diversity from simple units.

The authors differentiate between grammar-based and evolutionary Generative Design. In grammar-based designs, the principle of database amplification is exploited. The goal of grammar-based ideas is to generate complex forms and patterns from simple specifications, while evolutionary Generative Design approaches *'breed design solutions under the direction of a designer'*. Trying to define Generative Design on behalf of certain key properties, the paper suggests the following:

Firstly, the ability to generate complexity is necessary for design to be generative. Basic components are to generate aggregates of greater behavioral and structural complexity in the long run. Secondly, the complex and interconnected relationship between organisms and their environment is pointed out. The organisms adapt to their environment and their presence and number affects and changes the environment itself.

The next key property is the self-maintenance and self-reparation, i.e. the adaptation to maintain stable configurations within changing environments. You can make out an internal redundancy in their structure.

Lastly, the ability to generate 'novel' (meaning unseen and unpredictable) structures, behaviors, outcomes and relationships is crucial. Generative Designs must offer the potential to give rise to genuinely new properties that emerge during the process.

In his diploma thesis Jellinek and Masuch, Thomas Hitthaler presents the historical development of generative art/design in detail. For instance, he references John Maeda, who he calls a pioneer in this field, as he wrote one of the first programming languages to create simple designs generated by the algorithmic arrangement of points and lines on a computer. The process of programming and the aesthetics of the source code are as important as the resulting piece of art, says Maeda, which is a common

opinion among generative artists.

The author of the thesis suggests qualities on whose behalf one can judge Generative Design in order to compare it to classical design: firstly, you should pay attention to which possible output media there are as well as the quality of the result, obviously. Further, one should consider the range of functions the available tools have to offer and the usability of these tools. Generatively created designs are supposed to be editable, so Maeda, and the tools used to create them should be expandable.

According to Hitthaler's thesis, Maeda learned parts of his know-how on art and design, which he applied to the world of computers, from Paul Rand, who played a significant role in creating a basis of rules for design and could already be considered a generative thinker as he documented and verbalized his designs already, building a bridge between the fields of handcrafting, which had been following rules and mainly focusing on effectiveness and usability until then, and arts, which had until then usually not included obvious and explained rules given by the artist.

The challenge in generative, as in programmable, design, says Hitthaler, is that the ability to put your imaginations in formally exact and uniquely determined descriptions which a machine can translate, is necessarily needed.

2.2 Inspiration

To obtain the necessary familiarity with the programming language Processing, in which my program is written, several sources were helpful:

Above all, the book Bohnacker et al. [2009] provided me with plenty of inspiration with regard to programming techniques. It presents different ways to create natural and artsy designs with the help of randomness and noise. Works of different generative artists are introduced and interpreted and it provides the code for the reader to adjust parameters and apply small changes to easily create individual digital art. In the beginning of each chapter,

an idea for a technique that promises to create interesting and more importantly, diverse and extensive pieces of art, is presented and explained in its very basic idea. It's applied in more and more complex scenarios, sometimes extended into the third dimension and the chapters are concluded with so-called tools, with which you can draw choosing between a wide range of options. These include randomness and noise, oscillation figures, solids, attractors and several more.

Some of the pieces of art illustrated in the book are by Janne Kyttänen, a Finnish designer and conceptual artist, best known for his pioneering work with design for 3D printing. A lot of the inspiration that has gone into this work came from some of his works that were created in 2007. Those designs resemble natural structures, which is very much visible in his generatively created lamps. *Dahlia*, which is the name of the lamp first in figure 2.1, imitates the geometric arrangement of blossoms the dahlia flower presents. Next, the lamp Kyttänen called *610* bases on the fibonacci sequence, a recursive sequence that you can find in many natural patterns and clearly resembles a flower. Lastly, *Vasarely*, named after famous artist Victor Vasarely, shows a curvature out of a square.



Figure 2.1: top-left:Dah, below: Vas ; right: 610



Figure 2.2: more 3D-printed lamps by Kytäinen mor

The pattern visible in the Dahlia is one that is everywhere in nature and bases on the Fibonacci sequence, where one

Fibonacci numbers in nature

number is calculated by calculating the sum of the two before. Following this, it is easy to check that the first numbers are 0,1,1,2,3,5,8,13 etc. The function is a recursive one, namely $f(n+2) = f(n) + f(n+1)$. You can find this pattern in the 610, which resembles a cone flower, as well. In the following figure you can see the actual flowers and make out the resemblance. The reason why you can find the Fibonacci sequence in many natural patterns is that space efficiency is maximal using that pattern if you pick the right angle in which to place the high number of seeds. In his paper *The Sunflower Spiral and the Fibonacci Metric*, Segerman et al. [2010] at the Department of Mathematics of the University of Texas in Austin explains:

If you calculate the fraction of one Fibonacci number and the next, e.g. $3/5$ or $8/13$, you get close to the most rational number one can calculate, the so-called golden mean, and the bigger the Fibonacci numbers in the fraction get, the more precise is the estimation of that rational number. An approximation of it is 1.61803398875 and the corresponding angle, the golden angle, is 137.5 degrees. (You can calculate it by multiplying the non-integer part of the golden mean with 360 degrees and then taking its complement). With this angle, one obtains the optimal filling, that is, the same spacing between all the seeds, as one reaches the minimum of seeds laying in a way to exactly face each other. (2.4).



Figure 2.3: left:real Dahlia, right: real coneflower(610)

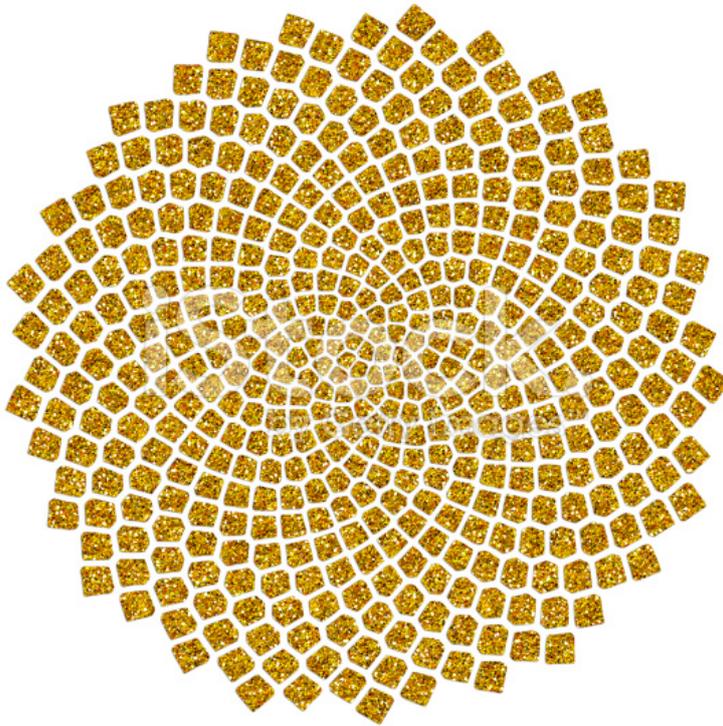


Figure 2.4: optimal spiraling with golden ratio

2.2.1 Nervous System

Similar looking lamps were designed by the Generative Design studio *Nervous System*. Their designs are inspired by natural phenomena as well and on top of that personalized in the design process. Experts in the fields of biology, architecture and mathematics, the two heads of the studio focus on complex, but natural structures, which can be translated into code. This way, their customers can watch their personalized products be created with the help of the online available applications. In figure 2.5 some of their designs in the field of illumination are shown.



Figure 2.5: generatively designed lamps by Nervous System

The first lamp as an example, is explained by its designers as follows:

The form is generated through a simulation of reaction-diffusion, a natural process that is theorized to be involved in everything from animal skin patterns to cell differentiation. For this lamp, we control the reaction through anisotropic diffusion. An-isotropic means varying the rate and direction of diffusion through space. This allows to create a form that is at once controlled and organic.



Figure 2.6: The design process for the 'reaction lamp' (first one in Fig. 2.5) ; Nervous System

Aside from lighting, the *Nervous System* designers also

turned their efforts into simulating plant growth and illustrating it in their applets as well as in their 3D models created with the help of laser technology. Most of their designs were written in Processing and some of their code is openly available for anyone to tinker with.



Figure 2.7: 'Laplacian Zoetrope' and 'Hyphae Zoetrope' ; Nervous System

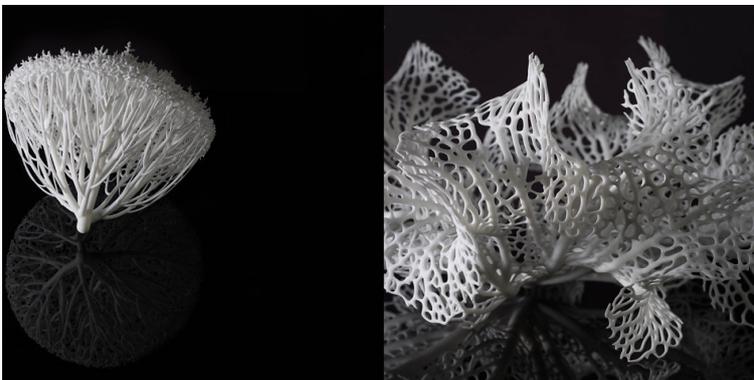


Figure 2.8: 'hyphae 3D 2' and 'hyphae crispata 1' ; Ner

The OBJexport library was created by *Nervous* to export Processing sketches for further 3D printing use. Having created the library for their own use (as in printing the products they sell), it was still shared with the community and used in my project as well.

2.2.2 Patrick Jouin

More 3D printed lamps imitating nature's design were offered by Patrick Jouin, whose lamps have the shape of flower petals and actually 'bloom': Figure 2.9 shows how one can open and close them interactively.



Figure 2.9: 'Bloom' ; Jouin [2010]

Materialise, a company dedicated to innovation with large covering in 3D printing as well, set up a *3D Printed Lamp Design Challenge for SketchUp* in 2010. In that context, more aesthetic flower shaped lamps were presented, as for instance the following two:



Figure 2.10: Exemplary designs for a 3D Printed Lamp Design Challenge Kurtsev [2010]

2.3 L-Systems in Nature

One can furthermore find out that the number of petals in many flowers is a fibonacci number, too.

On top of that, there is a lot of further explaining of natural phenomena which the discipline of mathematics has to offer. Considering the growth of trees or plants, one can clearly see that the growing happens in accordance with patterns, which led to the discovery of so-called L-Systems. Those are fractal and recursive systems, which allow to infinitely often repeat a pattern and get complex structures out of them.

There is a lot of literature on L-Systems, for instance the paper *Modeling plant growth and development* Prusinkiewicz [2004], in which Przemyslaw Prusinkiewicz from the Department of Computer Science at the University of Calgary presents virtual plants as 'computer models that recreate the structure and simulate the development'. It bases on one of Lindenmayer's papers (*Art and Science for Life: Designing and Growing Virtual Plants with L-Systems* Frijters and Lindenmayer [1974]) from 1974, in which L-Systems were introduced. Przemyslaw's paper was published in 2004, when possibilities were (a) accurately reproducing the structure and development of plants; (b) showing how architectural parameters affect the appearance of plants; (c) simulating plant physiology and investigating the effects of manipulations (e.g., pruning, bonsai techniques) or different external conditions (e.g. local light micro climate, water availability, crowding) on plant development; and (d) simulate plants not only in isolation, but also in their ecological contexts.

Where for Generative Art, the disciplines of computer science meet arts, the growing of virtual plants is an even more interdisciplinary area. Botany and applied plant sciences, mathematics, statistics, and theoretical computer science as well as computer graphics are all important and relevant in the creation of representative models. L-Systems are named after the biologist Lindenmayer who

first presented them in 1968.

An L-System grows according to production rules, which define replacements on the so called L-System alphabet, starting with basic axioms. As the systems are built recursively and with concrete rules, the little programming effort and the various modification possibilities make them appealing to programmers. As the study of interactions between plants and their environment is an important application of plant modeling, there are programs like *L-studio* and *vlab* to simulate such interactions to represent the plant and its environment.

With the help of his colleague Radomir Mech, the just mentioned Przemyslaw Prusinkiewicz also presented their own modeling framework to simulate and visualize interactions between plants and their environment. Figure 2.11 shows examples of how their models look like.

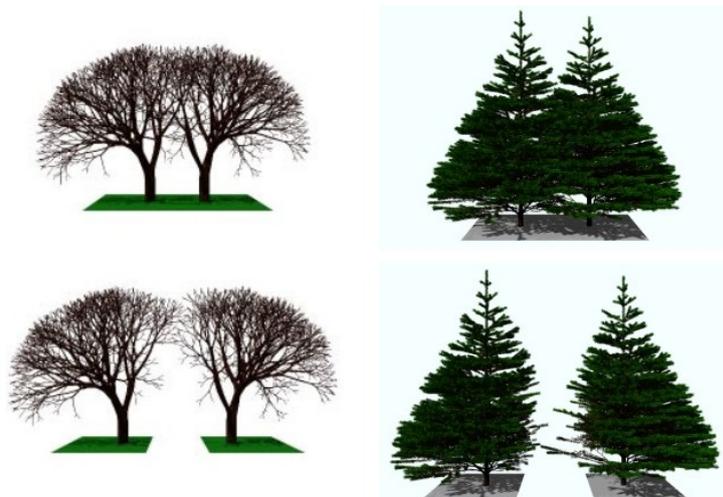


Figure 2.11: 2 models of trees competing for light ; Prusinkiewicz [2004]

2.4 The Superformula

More on the
superformula in
Chapter 4

In addition to the growth of plants, one can also pay attention to their shapes. Going further than just pointing at

the seed spirals and the number of petals, one can observe that spherical, spiral and cylindrical shapes are often observed in nature as well. Johan Gielis [2003] investigated this in his paper 'A generic transformation that unifies a wide range of natural and abstract shapes'. He points out that even though geometric morphometrics have been expanding, the "variability can be very large even within one species or genus". Also, while algorithms can yield perfect virtual plants, it is impossible to find an algorithm that exactly describes a real plant (Van Oystaeyen et al., 1996). In his paper, Gielis presents a generic formula, which can generate many abstract geometrical shapes that occur in nature. He shows that many geometrical forms can be interpreted as modified circles and calls these shapes super-shapes and the formula superformula. This formula was later extended to the third dimension. With the help of it, one can easily shape flower petals in three dimensions, as did in the following examples by the artists Stefano Tessarin and Stefano Tessarin taken from Borke [2003]:

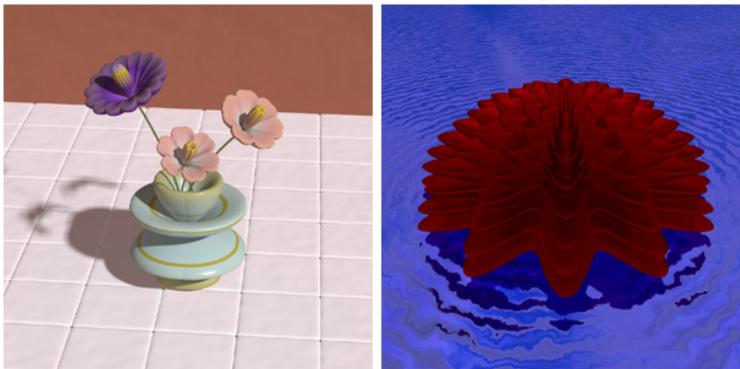


Figure 2.12: 3D models of flowery shapes created by the superformula

While in the artsy collections of designs created in order to 3D print, you find mostly inaccurate models that do not do justice to natural patterns, the very accurate models however are merely in order to represent and imitate the behavior of plants for research purposes. Therefore there is a lack of accuracy in some, a lack of aesthetics in other digitally created plants. The ones that offer the possibility of

3D printing are fixed designs, while the ones that offer interaction and animation are usually not printable.

2.5 OLEDs

Organic light sources
add to the natural
feel

The light sources used in this project are Philips Lumi-blades, which are one product in the *OLED technology* market. OLED stands for *organic light-emitting diode*.

Typically, they are used to create displays for screens of mobile devices, computers or TVs. Still, there has been profound research on using them for lighting purposes as they offer some advantages compared to common LEDs or LCDs: Working without backlights they can produce darkness most screens cannot reach. Further, in dark rooms their contrast ratio is higher than that of any LCD. Consisting of a highly flexible material, they offer interesting ways of lighting objects. Very thin and emitting "organic light" on their entire surface, they are the perfect light giver for LumiCAD.

2.6 Processing

Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals.

This is how the creators of Processing describe their language themselves. To be more precise, Processing is an extension of Java, which means that everything you can do in Java, you can also do in Processing. On top of that, Processing offers a visual component and functions to create sketches in a canvas to create digital art. It includes many

mathematical functions in the fields of geometry and many drawing and displaying functions similar to OpenGLs. One of the characteristics of Processing is its facility of use and user-friendliness. There is a lot of guidance for designers on how to use Processing for their purposes in form of video tutorials, long and comprehensive blog posts, books and the Processing forum, on which the community is fairly active.

2.7 Summary of Related Work

To summarize the result of this chapter, one can say that the topic of natural patterns and designing imitations of them is a very popular one. Yet, the following criteria will make the LumiCAD project differ from any other:

- offering interactivity and an easy-to-use environment for any kind of user
- using organic light to support the overall sentiment of nature
- offering the possibility to 3D print your result

Chapter 3

Own work

In this chapter, the reader will first be presented the basic concept this project was built upon. Interesting facts about the progress and results are presented. Moreover, difficulties and how they were dealt with are part of this chapter. Finally, one can see first final results by the end of it.

Starting out, there was the idea of a self-growing object, which would result in an aesthetic plant-alike product. First paper prototypes looked like the following:

First concepts

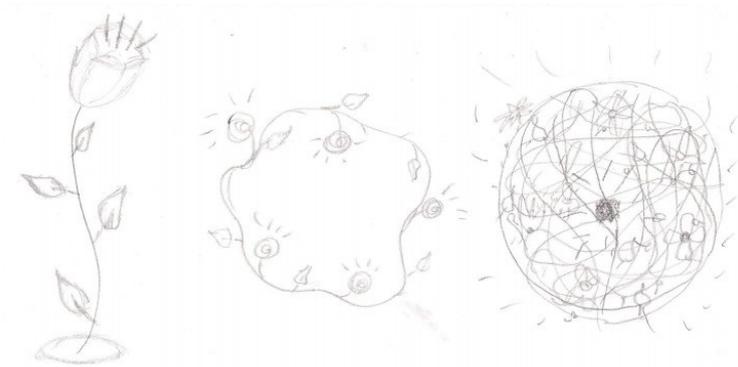


Figure 3.1: Paper sketches of the first idea

As one can easily distinguish, the three designs are different in a way that the dimension in which the growing

takes place adds by one: It can either grow to the top only, form a two-dimensional shape (e.g. a circle) or even a three-dimensional sphere. One can place one or more OLEDs to aim for different lighting effects: The one of a simple table/ standing lamp, or a wreath shining into many directions. Placing the light source in the sphere can create nice shadowing effects on the walls around it.

For the first case, the idea was to let a big bug create the main straw and put smaller ones on its sides for them to draw the leaves. Later the user should be able to pick a flower to put on top of it.

For the second case, the bug was supposed to draw the straw only, and the user to put leaves and flowers himself at the position of the bug by interacting (e.g. via the mouse).

Finally, for the three-dimensional case, I planned the interaction to work in a similar way as for the 2D one, but instead of one bug walking one circle, the user would be able to create many bugs walking in circles in a way that their joint ways would result in a path on an invisible sphere around the light source.

3.1 The growing process

3.1.1 Simple straws

A Bézier curve offers a way to create actually smooth and indefinitely scaled curves in computer graphics. Note that this is not trivial, as their parametric representations are not simple.

The main challenge for the simple flower lamps was the creation of aesthetic petals, which I will go into depth about in Section 2 of this chapter. Concerning the stalks, the approach was to use Beziér curves.

Beziér curves, as a short explanation, were derived from the so called *Bernstein polynomials*. They are defined as

$$B(t) = \sum \beta_i \cdot b_{i,n}(t) \quad (3.1)$$

with b being a Bernstein basis polynomial, i.e.

$$b_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i \quad (3.2)$$

You can create a Bézier curve of grade n by constructing intermediate points in between the starting point P_0 and P_n . The so-called Casteljau algorithm is used for easy calculation of Beziér points. The curve at point t_0 is evaluated as follows:

$$\beta_i^{(0)} = \beta_i \quad (3.3)$$

$$\beta_i^{(j)} = \beta_i^{(j-1)}(1 - t_0) + \beta_{i+1}^{(j-1)}t_0 \quad (3.4)$$

with $i = 0, n$ in 3.3 and $i = 0, n - j$ and $j = 1, n$ in 3.4.

Then, (t_0) equals to $\beta_0^{(n)}$.

Here you can see an example how (in this stage very roughly sketched) bugs create individually shaped and colored straws. The bugs are put in place by a mouse click and start walking from the position of the mouse.

In Processing, drawing a Beziér curve bit by bit is not offered by a predefined function, which is why the way to go to make the bugs walk on a Beziér path followed by a slowly growing straw was not a trivial task. The way to do it nonetheless was the use of `bezierPoint()`, a function offered by Processing to evaluate a Beziér curve at a point given four control points. Carefully randomizing those values but still keeping them in a range that guarantees growth in a sensible manner and calculating the `frameCount` modulo 1000 had the effect of the curve being evaluated at enough points to create a smooth movement of the bug. The next step was thus to let that bug create a path on its way.

Processing works in a way that there is a `setup()` function which is called once at the start of the application. The `draw()` function however is called once in every frame with a frame rate of 60 frames per second being the default setting. Slight changes in the created images with animations as an aim are therefore created through the code in the `draw()` function. Drawing an object at certain position coordinates which change in every frame will result in it "moving" over the canvas, if with every frame the background is drawn over the scene from the last frame and the next one will be added. Animating a growing straw can be reached by not updating the background with every call to

A short excursus
about the Processing
language

`draw()`, but instead keeping the same canvas and adding a bit of matter step by step. The combination of these two in form of a moving object creating a trace now creates the problem of having to clear the screen in every frame to the object is not drawn at all previous positions plus the current one, but also not being able to clear the screen without deleting the trace. The solution I found was to keep track of the previous positions with the help of an `ArrayList` of `PVectors`. One has to consider significant speed limitations with growing size of the path. In the first and second cases, those do not influence the speed of the program noticeably, yet in the three-dimensional case, in which one has to memorize the paths of all active bugs, they do.

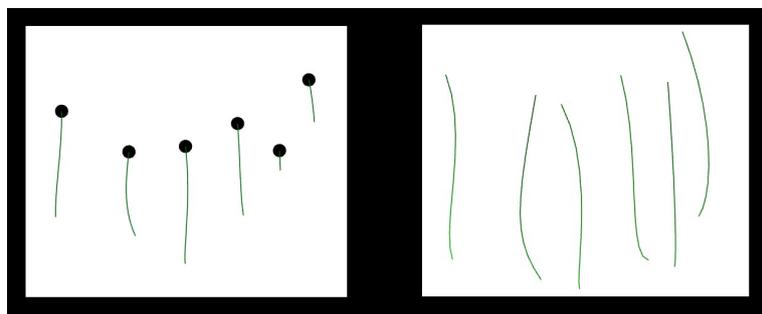


Figure 3.2: Growing straws in the first phases of the program

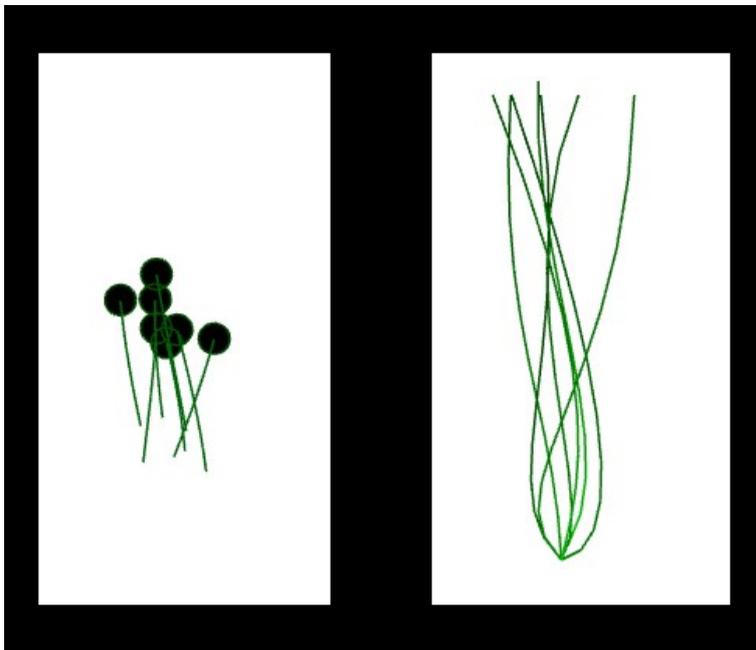


Figure 3.3: Combining straws to a bouquet

3.1.2 Creating Wreaths

In order to let the bugs walk in a circular manner, I chose to make use of trigonometric functions defining a circle (i.e. the sine and cosine):

To create a circle, one has to iterate over the angle and calculate the cosine of it for the x position, and the sine for the y position to then multiply with the desired radius. The circle is full when the angle reaches 360 degrees, which is the condition for the loop to stop. This way, one can let the bug walk along the path and create a wreath behind it:

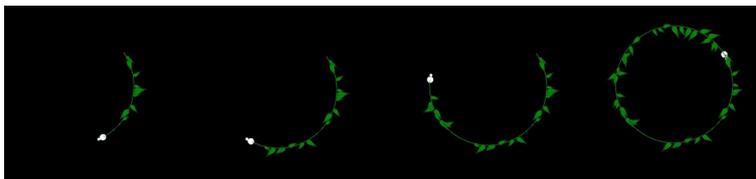


Figure 3.4: Bug creating a wreath

The first bits of interactivity the program will offer by the end of the implementation are implemented in this version already: The user initiates the growth of leaves with the mouse while the bug walks; this way the leaves grow at the last position of the bug. Obviously, he has the possibility to create petals too, as those carry the light sources in the lamp. For this purpose, a leaf class and a flower class contain the functionality to draw those. Saved in an array which is updated with every mouse click, all active leaves are drawn in every frame.

3.1.3 Getting More Complex

To expand the model into three dimensions, one needs to add a z component to the trigonometry of the shape to draw. Therefore spherical coordinates ϕ and θ are used to calculate the Cartesian coordinates:

$$\begin{aligned}x &= \cos(\theta) \cdot \sin(\phi) \cdot \text{radius}; \\y &= \sin(\theta) \cdot \sin(\phi) \cdot \text{radius}; \\z &= \cos(\phi) \cdot \text{radius}\end{aligned}$$

To generate different unique patterns, the updating of the coordinates does not happen consistently, but with the help of randomization. The user puts bugs into the scene and lets them walk their individual paths, yet he can stop the path from being drawn any further by killing the assigned bug or remove the entire path. This procedure results in straws around an invisible sphere. The position of the light source is yet to determine, following the most intuitive idea, one could put them into some of the petals as in the two-dimensional case. An alternative however is to mount a light source, for instance a cube made out of OLEDs, into the middle of the sphere to lighten the structure from inside.

Moreover, the paths actually walked by the bugs fol-

low randomized patterns to ensure that they stay on the spherical basis and still create diversity.

To make it possible for numerous bugs to walk around individually on the screen and be interacted with each by its own, a class was created for them. To kill precisely the bug one chooses, the color of the pixel clicked on is compared with the uniquely determined colors of the bugs and if it is equal to one of them, the bug dies. As the entire scene is drawn from scratch in every frame (see above), one can simply exclude a path clicked upon with the right mouse button from being drawn in the next frame.

3.2 The Flower

The development process of creating a realistic and aesthetic design of a mesh to represent the petals of a flower was full of challenges. The original idea had been to create these in a 3D modelling software to then import them into Processing. However, the functionality of the library for that task did not fulfill the requirements to import such a mesh, but merely simple and small ones. To find an alternative, I explored the built-in possibilities Processing has to offer.

3.2.1 Import Problems

Firstly, I tried to use the two libraries for such an import listed on the Processing site (MRI3DS and ObjImport), which are by Victor Martins (Martins [2010]). It was cumbersome and time consuming to get his own examples to work with the library and impossible to get it to work with more complex own examples. Similar experiences resulted from using all other libraries existing for this purpose, which is why after losing a considerable amount of time, I decided to drop the idea of importing externally modeled files into my sketches, which was quite a let down and led to significant changes in the original LumiCAD concept.

see Future Work for more information

3.2.2 Non-Basic 3D Shapes in Processing

To create meshes in Processing, one has to draw the mesh by providing the coordinates of all vertices the mesh consists of. One way to do that is to find a mathematical representation of the relationship between the individual vertices, as one would do to create a sphere without using the built-in function.

It is obvious however that the amount of shapes one can create in this way is limited by the possibilities to mathematically express a shape. As mentioned in the previous chapter, so-called *supershapes* claim to represent shapes that can be found in nature. Experimenting with values led to the shape you can see in Figure 3.5 following a brief explanation of the formula.

The superformula creates all kinds of shapes occurring in nature, e.g. flowers

The superformula - as a quick reminder - is a geometrical approach to model natural shapes and reads

$$r(\phi) = \frac{1}{\sqrt[n_1]{\left(\left|\frac{1}{a} \cdot \cos\left(\phi \cdot \frac{m}{4}\right)\right|\right)^{n_2} + \left(\left|\frac{1}{b} \cdot \sin\left(\phi \cdot \frac{m}{4}\right)\right|\right)^{n_3}}} \quad (3.5)$$

In his paper, Gielis first reminds of papers from Loria, 1910 and Gridgeman, 1970, whose work resulted in the definition of so-called *superellipses*, which are a generalization of a set of shapes containing circles, squares, ellipses and rectangles. The formula of those superellipses is

$$|x/a|^n + |y/b|^n = 1 \quad (3.6)$$

Criticizing the disadvantage of the limit of symmetry in superellipses, he motivates the creation of the superformula. For $n_1 = n_2 = n_3 = 1$, the formula forms an ellipse and if $a = b = 1$, it yields a circle. The variable m states how many points are fixed on the unit circle and influences their spacing. The plane is divided into a number of sectors equal to m . The values of n_2 and n_3 determine whether the

shape is inscribed or circumscribed in the unit circle. The value of $n1$ further determines the shape. For a list of plants whose shapes can precisely be described with this formula, a read of Gielis [2003] paper is highly recommended. In it, he explains the deduction of the formula in all detail and refers to many other interesting results of the interplay of mathematics in nature. Moreover, it is important in this context to mention that it is said to be possible to obtain the parameters for the formula given a certain shape. This might make it possible to scan flower shapes occurring in the real world, get the parameters and exactly duplicating the shapes even within Processing to create a wide range of different shapes.

To expand it into higher dimensions, one can calculate the spherical product of superformulas. The coordinates of the 3D version is therefore obtained by calculating

$$\begin{aligned}x &= r_1(\theta) \cdot \cos(\theta) \cdot r_2(\phi) \cdot \cos(\phi) \\y &= r_1(\theta) \cdot \sin(\theta) \cdot r_2(\phi) \cdot \cos(\phi) \\z &= r_2(\phi) \cdot \sin(\phi)\end{aligned}$$

where ϕ stands for the latitude, θ for the longitude and both range from $-\pi$ to π .

The following images show the results of experimenting with the parameters of the superformula to end up with shapes resembling flowers within the supershapes creator written by bitcraft:

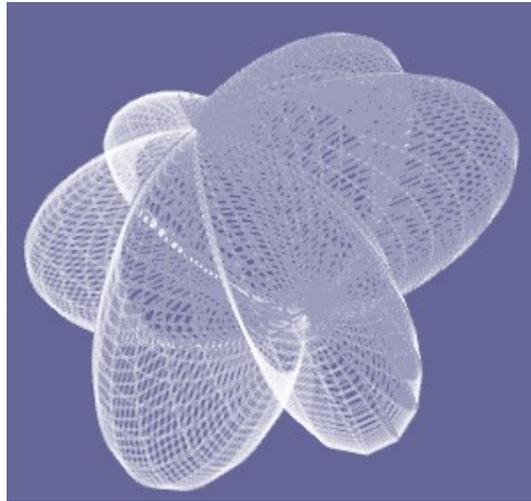


Figure 3.5: Superformula creating a supershape resembling a flower

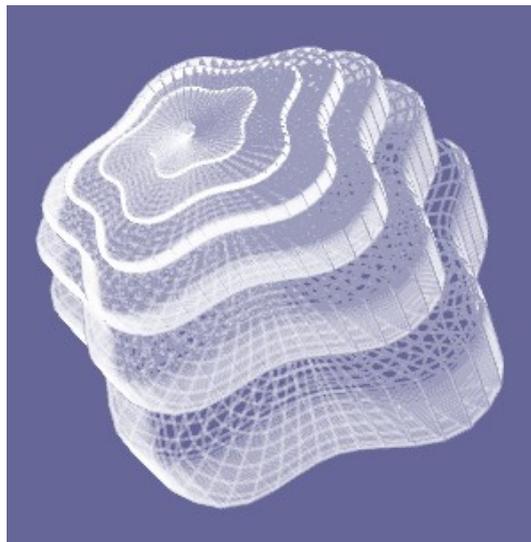


Figure 3.6: Superformula creating another supershape resembling a flower

I further worked on creating an own triangle mesh that would make a sunflower to experiment how one can imitate natural shapes using simple primitives as triangles instead of complex formulas, just a collection of vertices. The result is shown in Figure 3.7.

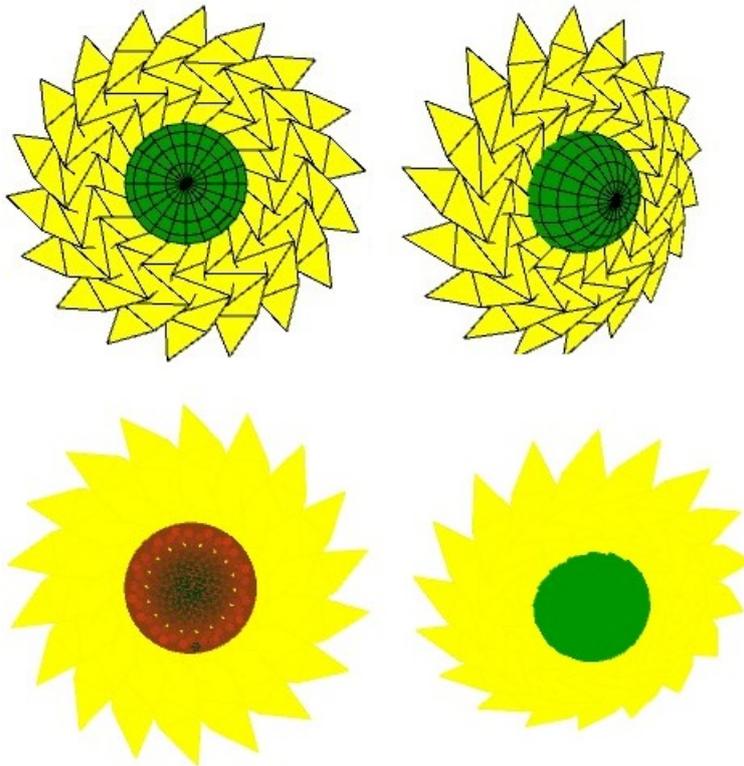


Figure 3.7: "Drawing" a sunflower

A wreath created with this kind of flower looked like this as an OBJ Model (note that the holes are supposed to be placeholders for the OLED tiles to be placed in the center of the flowers):



Figure 3.8: Finished model of a lamp ready to 3D print

One can imagine to let this kind of lamp hang from the ceiling and putting light sources into the three sunflower models. As a possible size one could think of around 15cm diameter for one sunflower, thus a wreath with the radius of approximately half a meter would be desirable. One can however scale it to any desired size and use OLEDs of according sizes.

Randomly created three-dimensional shapes created with the superformula flower resulted in the following:

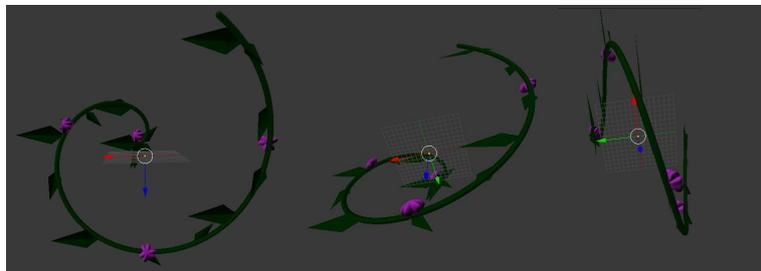


Figure 3.9: 3D model from different perspectives

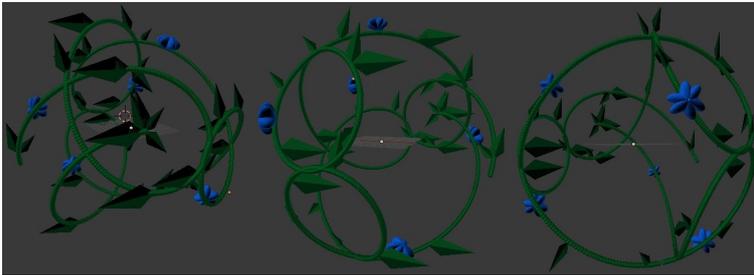


Figure 3.10: another 3D model from different perspectives

Note that all pictures in one row depict the same model. The application just before exporting these two models looked like the following:

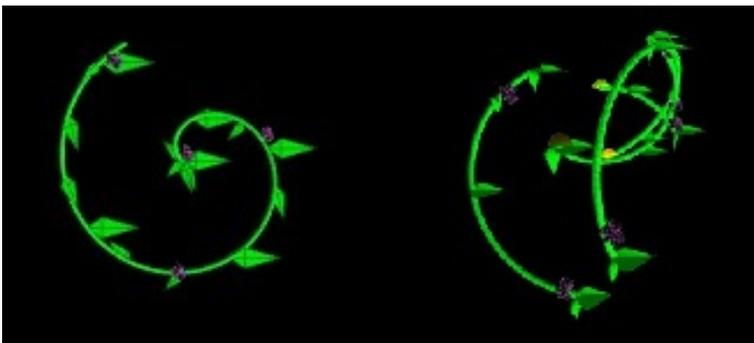


Figure 3.11: Screen shots taken of the designing environment before exporting the models from above

3.3 The Process of Printing the Model

3.3.1 Leaves

As a last step it was intended to convert the exported OBJ file to an STL file and simply print the result. This, however, did go about without difficulties. Firstly, the OBJ file exported in the first tries did not include the leaves that you can see in previous screen shots of the image in Processing. The problem seemed to be that the curved shape of them

was causing problems which I could not fix in the given time span, which is why their shape changed to a more triangular, artificial shape. Picture 3.12 shows how the shape changed and one can observe that the difference is not so obvious in the use in a packed scene anymore:

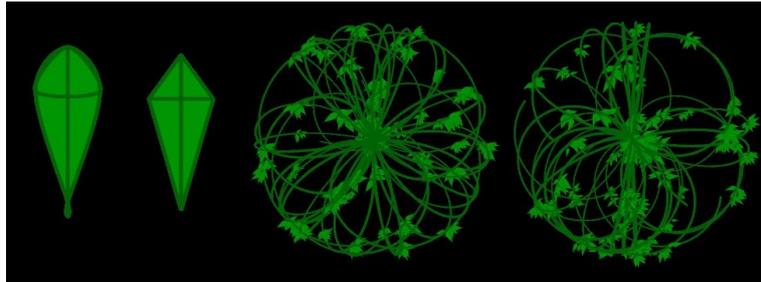


Figure 3.12: Before vs after: one leaf and plants with the respective leaves

However, the leaves continued to cause trouble. Even though the new leaves were visible in the OBJ file after the export, in the last stage, in which the actual printing was supposed to happen, the software of the 3D printer did not process the leaves that had been visible in the STL file properly, but only sparsely and weirdly transformed.

Chapter 4

Evaluation

4.1 System Usability Scale

I used the software system the System Usability Scale (SUS) [Brooke, 1996] to evaluate my program. It consists of a questionnaire covering ten questions with five response options from *Strongly agree* to *Strongly disagree*. Originally created by John Brooke in 1986, it was used frequently proving its quality.

The statements are either posed positively or obviously negatively. By having similar questions the participant is motivated to reconsider his prior choice that was maybe not too well-thought-through. The positive and negative questions alternate so that an optimally filled sheet would show the typical zick-zack pattern of 1s and 5s. The score is calculated by subtracting the score of the negative questions from 5 and decrementing the scores of the negative questions by one. This way, one gets scores from 0 to 4 where 0 is the worst and 4 the best result. Multiplying the sum of values calculated in this way for every user with 2.5 results in a number between 1 and 100. The average gives a measure for the quality.

The System Usability Scale quantifies the usability of a system.

4.2 User Study: Part I

In a first questionnaire, participants were asked about their personal background and experience with design and programming. The goal was to have a diverse range of people participating, as the program is supposed to be used by anyone and not only by designers or programmers. 10 randomly picked students were asked to provide their genders, ages, and field of study and then to answer some questions about designing and personal fabrication.

4.2.1 Results of the background-based questions

Participants of the study were students of the fields of Biology, Architecture, Computer Science, Social Sciences, and Civil Engineering. Their ages ranged from 21 to 25 and six of them were female.

Their answers suggested that most of them (7) had no prior knowledge of programming and half of them had not designed anything themselves yet. None of those 5 students had fabricated anything designed yet and three more had designed things, but never actually fabricated their designs. Nonetheless, 7 participants claimed that personal fabrication appealed to them and that they would consider it fun to fabricate their own products in easy ways. 7 participants considered their lamps aesthetic.

As a result one can say that personal fabrication remains to be something that sounds appealing to many, but was not explored by most participants. On top of that, the idea of the program itself seemed to be liked and the achieved results have pleased the majority of the users.

4.2.2 Results of the questions taken from the SUS

The SUS score the system achieved was 81.25, which is fairly good considering that the scale goes from 0 to 100. You can see the exact results for all the participants in Table 4.1.

It is obvious that most participants answered alike. For instance, most participants did not plan to use the system too frequently, which is understandable as its concept does not suggest it: not many average people would 3D print a new lamp on a regular basis. Very similar answers came for the statement "I think that I would need the support of a technical person to be able to use this system" as well, as it is obviously not a program that requires any guidance other than the one provided within the program itself. Very different results came up for the statements "I found the various functions in this system were well integrated", as it seemed to cause confusion as what it meant for functions to be well integrated for some participants.

| Participant | SUS score |
|-------------|-----------|
| P1 | 85.0 |
| P2 | 85.0 |
| P3 | 80.0 |
| P4 | 72.5 |
| P5 | 82.5 |
| P6 | 87.5 |
| P7 | 82.5 |
| P8 | 82.5 |
| P9 | 77.5 |
| P10 | 77.5 |

Table 4.1: System Usability Scale results

Chapter 5

Summary

5.1 Summary

5.1.1 One final result

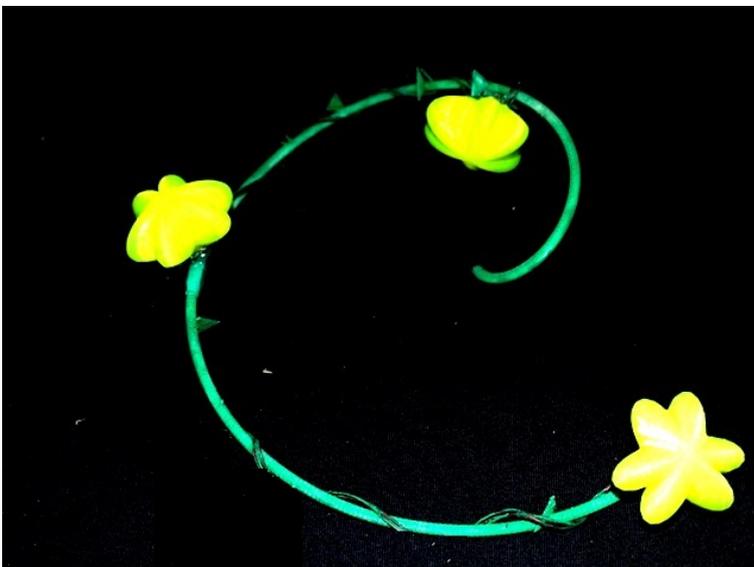


Figure 5.1: Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs



Figure 5.2: Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs



Figure 5.3: Photograph of the final model. Note it is not a real lamp as printed with the program but only a model illuminated with LEDs

The model shown in the previous figures resulted from restrictions in time and resources. As for this purpose, only a small printer was available, it was not possible to print the model in a large enough size to fit OLEDs into the petals. Therefore for demonstration purposes, they were illuminated by simple LEDs, which also explains why cables attached to the main straw are visible. In the real lamp printed in ideal circumstances, the flowers will be illuminated each individually and attached to the straw already, while in the model they had to be printed separately and attached to the separately printed straw for reasons of only having a printer available that could print in one color.

5.1.2 Lessons learned

As one lesson one can clearly say that Processing is not the right environment to easily fabricate aesthetic items oneself. It takes expert knowledge to create complex meshes and the use of poorly functioning libraries to export those and there is generally very little help and guidance in cases of troubleshooting. However, I believe that the ability to create personal and artsy fabrics with the help of code can result in satisfying results if one invests enough time in studying programming, computer graphics and geometry. The goal of making the personal fabrication interactive is still not reached in a level in full freedom of creation for the user and remains to be worked on.

In addition, the interactive program I wrote for this thesis demands unavoidably high computation times as Processing is not suited for interactivity in 3D scenes yet.

To later improve the look and feel of the prototype result, I used Blender for post-processing. It proved to be considerably better suited for tasks involving basic transformations and interactive editing than Processing as one gets immediate results of ones actions instead of having to compile the code after every change to see the effect, which is why one could certainly think of creating a program on the basis of Blender for everyday type of people to create 3D models.

5.1.3 Some Things to Note

The concept of being able to "create a lamp oneself" with the given tool obviously requires access to a computer with the Processing language installed on it, but furthermore other restrictions came up that had not all been anticipated at the outset:

A 3D printer with the ability to print in different colors is needed to print the lamp exactly as it looks in the preview (note that color printing is not yet to be taken for granted as a main feature of 3D printers). Even with color printing, the colors available are usually not as diverse and available in every shade the RGB code can create, which is why the

printed model will always look slightly less natural than it does on the screen.

5.2 Future Work

There are numerous things that presented themselves as future work fields. Obviously one can add more and more flower options and different kinds of leaves, maybe different types of straws (for instance, wood-like) and more algorithms for the flower to grow. Instead of growing a flower in the first option, one can consider growing a tree (even though it is questionable the result would make an aesthetic lamp).

For the two-dimensional case, one can imagine the bug walking not in a perfectly circular manner, but rather with slings and snares and in elliptic, more rectangular-like or really any kind of way. For this, a further use of the superformula could be helpful. As it is in art, there are endless possibilities one can explore.

In the 2 and 3D cases one can consider putting different kinds flowers into the same plant, in the 1D case one could create a bouquet of different flowers.

On top of that, as mentioned above, I unfortunately had to change the original design of leaves with curved vertices to ones one could create using a triangle mesh, as the OBJExport library was not capable of including the original leaves into exported OBJ files. For all shapes that are not basic 3D primitives as offered by processing or triangle or quad strips, the library failed to correctly fulfill its purpose. One can hope for the development of a more powerful and capable library to export 3D models from Processing, which one could then use for future work on this project.

With newer, deformable or simply smaller OLEDs one can explore how it would look like to integrate them into the matter and let the entire plant shine from inside through slightly transparent material. As always with 3D printing, one could also try different materials to find one that looks

most natural and organic.

To solve the problems of difficulty in coding complex shapes in Processing, one can hope for the generation of a functioning import library to then import 3D models created in other environments (e.g. Blender) into the program and integrate them into the sketch. This way, it would be very easy to offer a large repertory of blossoms and leaves and therefore a more interactive pick and put environment.

As mentioned in Chapter 3 already, it would most certainly make sense to further explore how one can obtain the parameters for the superformula given a real flower to then easily create very correct imitations of all flower shapes in Processing. However, throughout my research I have not encountered anyone documentedly working on this.

To increase interactivity, one would want to aim at adding the option to select a part of the sketch (e.g. a flower or leaf) and manually increase/ decrease/ rotate/ move those interactively, yet I am, due to my experiences with this project, very certain that this would blast Processing's capabilities and would result in a heavy, inefficient product. Similarly, it would be desirable to have sliders to change the color tone of certain pieces of the sketch (especially the blossoms). Displaywise one could add lighting effects for the user to an imitation of the lighting effect the lamp would have in real life even before printing the lamp. It is in my opinion in any case worth exploring alternatives to Processing to reach for this goal.

One more idea that occurred to me was expanding the naturality of the plant growth by including a virtual sun, let-

ting the user make it rain and maybe prolong the creation process in an imitation of real plant growth.

Appendix A

Evaluation Details

A.1 Exact format and results of the SUS user study

The following pages show the sheets that were handed out to the participants of the user study and a summary of the results:

USER STUDY - LumiCAD

Personal and background information

I am female male no indication

My field of study _____

Age _____

Agreement

++ + 0 - --

I think it would be fun to be able to design everyday objects myself.

I am familiar with programming.

I already have experiences in designing.

I haven't designed anything yet but would want to if it was easy to do so.

I think that in future a lot of personal designing and fabrication will happen.

I have experience with 3d modeling software.

I think that the lamp I created is aesthetic.

Personal fabrication appeals to me as something I would like to do.

System Usability Scale

© Digital Equipment Corporation, 1986.

| | Strongly disagree | | | | | | Strongly agree |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--|----------------|
| 1. I think that I would like to use this system frequently | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 2. I found the system unnecessarily complex | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 3. I thought the system was easy to use | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 4. I think that I would need the support of a technical person to be able to use this system | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 5. I found the various functions in this system were well integrated | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 6. I thought there was too much inconsistency in this system | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 7. I would imagine that most people would learn to use this system very quickly | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 8. I found the system very cumbersome to use | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 9. I felt very confident using the system | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |
| 10. I needed to learn a lot of things before I could get going with this system | <input type="checkbox"/> | | |
| | 1 | 2 | 3 | 4 | 5 | | |

RESULTS SUS

Results on personal and background information of the participants

I. Personal

| | Gender | Age | Field of Study |
|-----|--------|-----|-------------------|
| P1 | female | 22 | Biology |
| P2 | female | 23 | Architecture |
| P3 | female | 22 | Architecture |
| P4 | male | 22 | Computer Science |
| P5 | male | 21 | Computer Science |
| P6 | female | 24 | Pharmacy |
| P7 | male | 23 | Civil Engineering |
| P8 | female | 20 | Civil Engineering |
| P9 | female | 23 | Social Sciences |
| P10 | male | 22 | Social Sciences |

II. Questionnaire results (numbers = #participants with the respective answer)

| Agreement | ++ | + | o | - | -- |
|---|----|---|---|---|----|
| I think it would be fun to be able to design everyday objects myself. | 6 | 4 | | | |
| I have experience with 3d modeling software. | 2 | 1 | | | 7 |
| I already have experiences in designing. | 3 | 2 | | 4 | 1 |
| I haven't designed anything yet but would want to if it was easy to do so. | 6 | 3 | 1 | | |
| I think that in future a lot of personal designing and fabrication will happen. | 5 | 4 | 1 | | |
| I have actually fabricated an own design at least once. | 3 | | | | 7 |
| I think that the lamp I created is aesthetic. | 4 | 4 | 2 | | |
| Personal fabrication appeals to me as something I would like to do. | 7 | 3 | | | |

SUS scores of the participants

| Question | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | TOTAL |
|----------|---|---|---|---|---|---|---|---|---|----|-------|
| P1 | 2 | 1 | 5 | 1 | 3 | 1 | 5 | 1 | 5 | 3 | 85 |
| P2 | 3 | 1 | 4 | 1 | 4 | 1 | 5 | 1 | 4 | 2 | 85 |
| P3 | 1 | 1 | 5 | 1 | 3 | 3 | 5 | 1 | 5 | 2 | 80 |
| P4 | 4 | 2 | 4 | 1 | 3 | 3 | 4 | 1 | 3 | 2 | 72.5 |
| P5 | 2 | 1 | 5 | 1 | 4 | 2 | 5 | 2 | 4 | 1 | 82.5 |
| P6 | 3 | 1 | 5 | 1 | 3 | 1 | 5 | 1 | 4 | 1 | 87.5 |
| P7 | 1 | 1 | 5 | 1 | 4 | 1 | 4 | 1 | 5 | 2 | 82.5 |
| P8 | 3 | 1 | 4 | 1 | 5 | 1 | 4 | 2 | 3 | 1 | 82.5 |
| P9 | 2 | 1 | 5 | 1 | 3 | 3 | 5 | 1 | 4 | 2 | 77.5 |
| P10 | 1 | 2 | 4 | 1 | 4 | 2 | 4 | 1 | 5 | 1 | 77.5 |

Average: 81.25

Bibliography

Online, accessed July 13, 2015. URL <http://www.project05.com/2010/04/610-table-light-by-janne-kyttanen/>.

Online, accessed July 13, 2015. URL http://gewerbemuseum.ch/fileadmin/user_upload/Gewerbemuseum/Ausstellungen/pressematerial/Pressematerial_2002_bis_2010/KunstStoff/230_FOC_Dahlia2.jpg.

Online, accessed July 13, 2015. URL <http://www.jannekyttanen.com/#modal8351616142>.

Online, accessed July 13, 2015. URL http://www.within4walls.co.uk/view_info.php?id=MGX+by+MATERIALIZER.

bitcraft. Super duper shape explorer. Online, accessed July 11, 2015. URL <http://openprocessing.org/user/1720>.

Hartmut Bohnacker, Benedikt Groß, Julia Laub, Claudius Lazzeroni, Benedikt Gross, Julia Laub, and Claudius Lazzeroni. *Generative Gestaltung: entwerfen, programmieren, visualisieren*. Schmidt Mainz, 2009.

Paul Borke. Supershapes/ superformula, 2003. Online, accessed June 21, 2015. URL <http://paulbourke.net/geometry/supershapes/>.

John Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

- D Frijters and A Lindenmayer. Art and science for life: Designing and growing virtual plants with l-systems, 1974.
- Johan Gielis. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American journal of botany*, 90(3):333–338, 2003.
- Dipl Ing Brigitte Jellinek and Dipl-Ing FH Claudius Masuch. Generative erzeugung von design mit vvvv.
- Patrick Jouin. Bloom table lamp designed by patrick jouin, 2010. Online, accessed June 21, 2015. URL <http://www.mgxbymaterialise.com/limited-editions/mgxmodel/detail/detail/71>.
- Oleg Kurtsev. Equations by oleg kurtsev, 2010. Online, accessed June 21, 2015. URL <http://dev-blog.sketchup.com>.
- Victor Martins. Libraries for java and processing, 2010. Online, accessed July 11, 2015. URL <http://www.pixelnerve.com/processing/libraries/>.
- Jon McCormack, Alan Dorin, and Troy Innocent. Generative design: a paradigm for design research. *Proceedings of Futureground, Design Research Society, Melbourne*, 2004.
- Przemyslaw Prusinkiewicz. Modeling plant growth and development. *Current opinion in plant biology*, 7(1):79–83, 2004.
- Henry Segerman et al. The sunflower spiral and the fibonacci metric. In *Proceedings of Bridges 2010: Mathematics, Music, Art, Architecture, Culture*, pages 483–486. Tessellations Publishing, 2010.

Index

| | |
|------------------------------|-------------------------|
| abbrv | <i>see</i> abbreviation |
| Bernstein polynomials | 20 |
| Beziér curves | 20 |
| Evaluation | 35 |
| evaluation | 33 |
| Future Work | 41–43 |
| OLED | 16 |
| Related Work | 3 |
| Superellipses | 26 |
| Superformula | 26 |
| System Usability Scale | 33 |

