**Chapter 5**

**Codeable Ob jects**



Codeable Objects is computational design tool that allows people to design and export the toolpaths for a laser cut lamp. Lamps possess an established function, but can vary immensely in their form-factor. As a result, the domain of lamp design offers a wide set of design possibilities for a concrete and useful end product. By introducing lamp design in the context of algorithmic craft, my

goal was to allow people to construct unique lamps comprised of computationally generated forms and patterns.

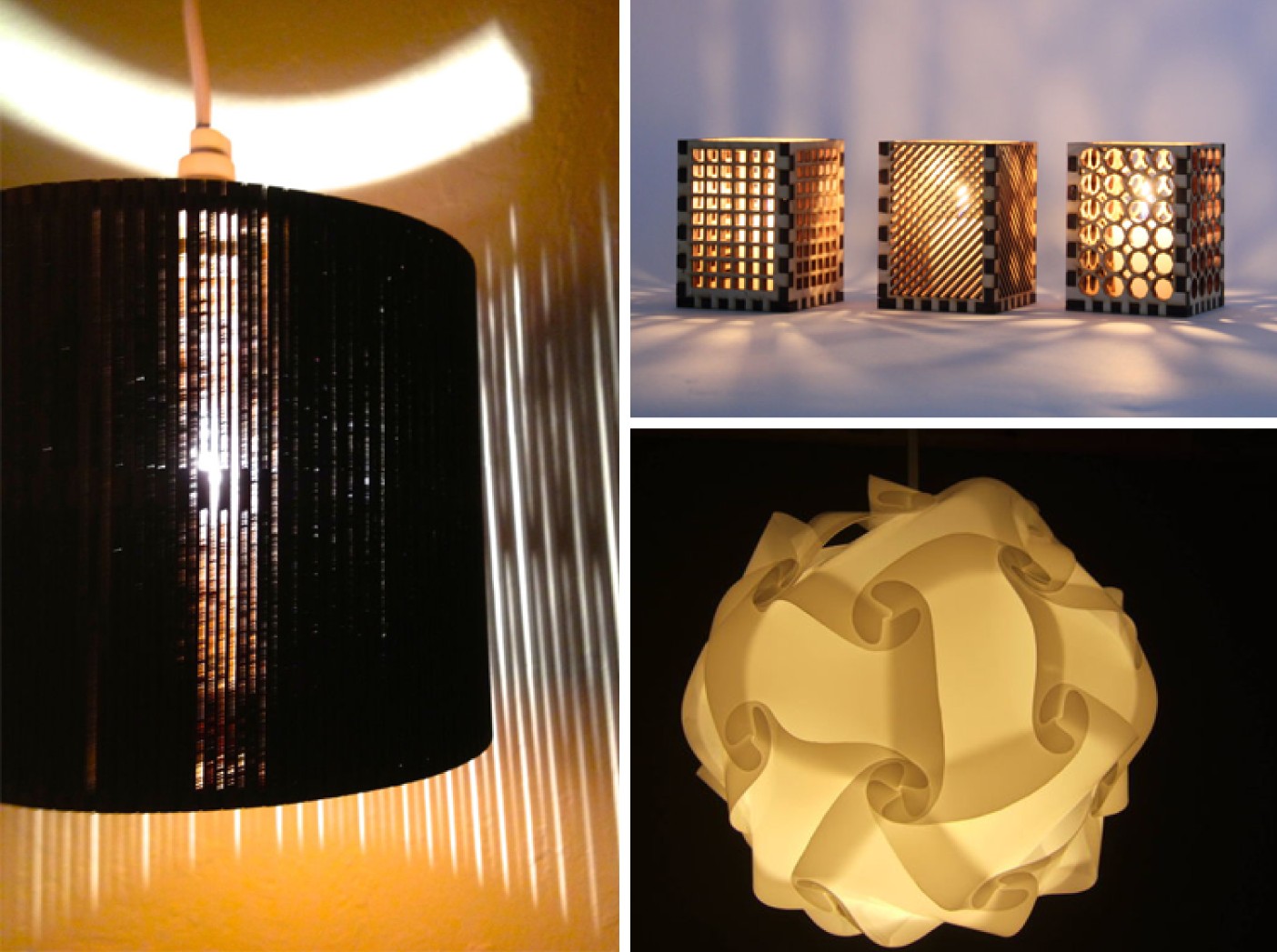


Figure 5-1: a selection of laser cut lamps from Instructables

**5.1 Motivation**

There is a precedent for creating DIY lamps through digital fabrication. The Instructables commu- nity tutorial website has an entire section devoted to DIY lamps, and many examples of patterns that use a laser cutter for fabrication. Many of these Instructables contain minimal design flexibility; they provide instructions that allow people to reproduce the lamp in the tutorial, but do not describe methods of deviating from the original design. The tutorials that do encourage design flexibility often require the person creating the lamp to use professional CAD tools to create their own design. In one popular laser-cut-lamp Instructable, the author recommends using Solidworks to design the form and Illustrator to create the pattern on the shade [24]. SolidWorks is more difficult to use than Illustrator, but has the benefit of being parametric. Conversely, Illustrator contains support generating aesthetic forms and patterns and is well suited to creating individual illustrations for

laser cutting, but lacks the parametric functionality needed for the design of artifacts with multiple parts. Both tools are extremely challenging for first-time users.

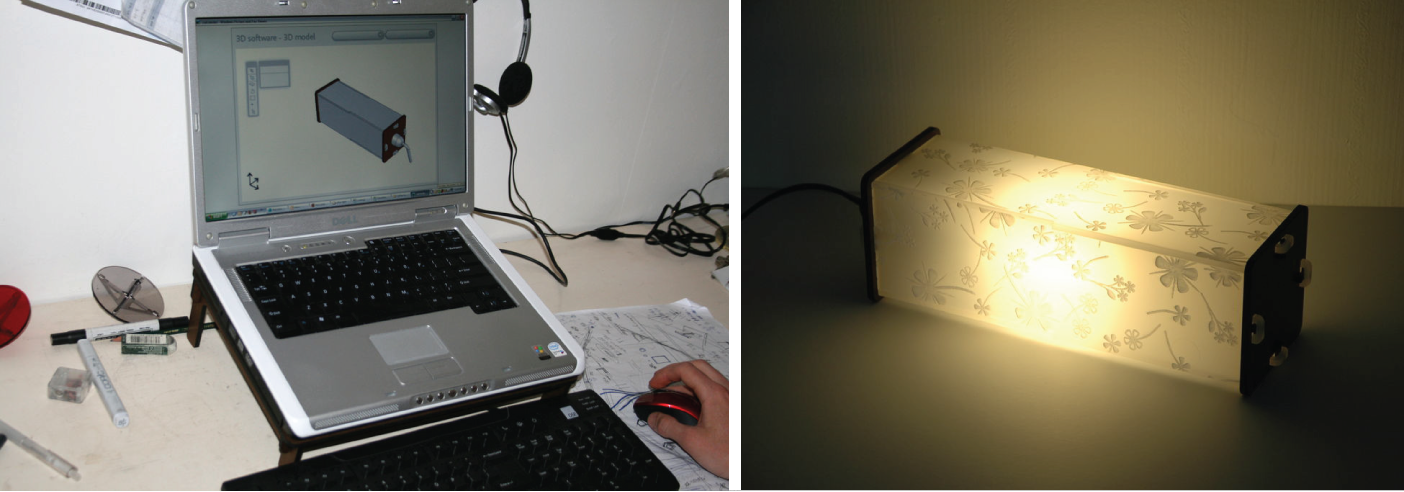


Figure 5-2: Instructables lamp tutorial with SolidWorks design process

Using existing software tools for lamp design also presents other challenges. One common way to make laser-cut 3D forms is by assembling 2D press fit pieces. I found that when creating 3D forms that were curved, it was extremely challenging in 2D-CAD software to correctly size and design parts which would fit the faces of the lamp frame, creating the lamp shade. Although it is possible to create lamps from bare laser cut frames, the shades themselves provided an excellent space for incorporating aesthetic illustrative elements. Without parametric functionality however, it is difficult to modify the aesthetics and form of the lamp in a back-and-forth manner. Computational design offered a solution to many of these problems, while simultaneously providing a form of constructive engagement in programming, fabrication and craft.

**5.2 Tool Description and workflow**

The first version of Codeable Objects attempted to combine parametric manipulation, aesthetic pattern generation and the conversion of a 3D form to 2D parts into a single computational design tool for lamp design. The lamp itself was comprised of five basic parts, a wooden press fit frame, a set of vellum pieces that fit over the frame to act as a shade, a set of cardstock pieces with a pattern that fit over the shades, and a pre-made standard light fixture that fit into the frame (see figure:

5-3.)

Codeable Objects was developed as a programming library for Processing and contained a set of programming methods that allow the user to describe the lamp, and define the tool paths for all three materials. In the first version of the library, all design took place via textual programming

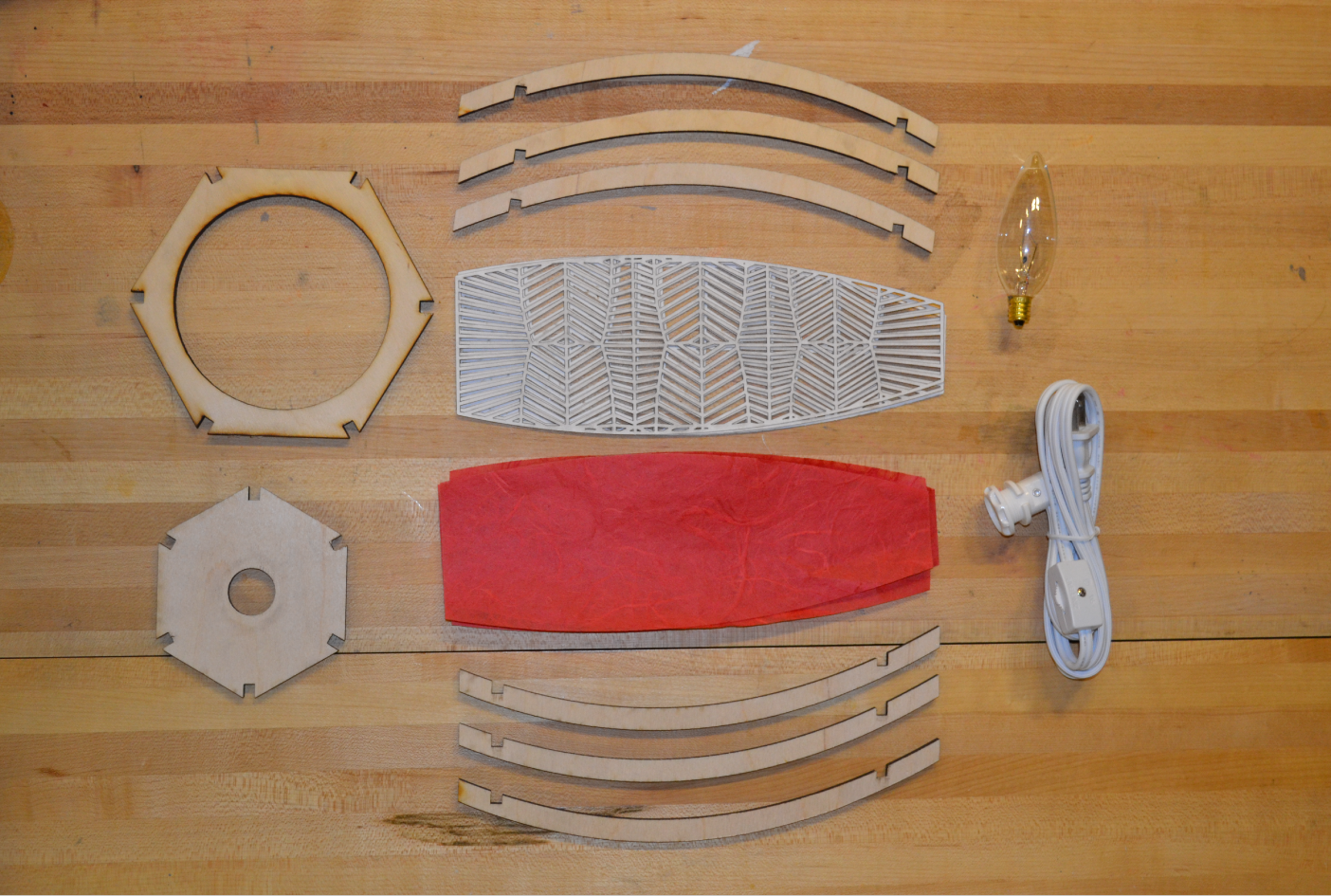


Figure 5-3: the individual parts of a lamp

and keyboard commands. There are four main functions in the Codeable that determine the height, top width, middle width and bottom width of the lamp. The library also provided access to an additional set of methods that control the number of sides, the resolution of the curve and the position of the internal structural supports. To facilitate the construction process, notches are automatically generated in all of the parts to allow the lamp to be press-fit together. The shades are also automatically generated to fit the form specified by the user. The inclusion of this feature gives the user freedom to customize the shape of their lamp, without having to worry about the mechanics of construction.

Codeable Objects also includes a second set of programming methods that allow users to describe the decorative components of the lamp by specifying coordinates in polar or cartesian space. Upon compilation, the coordinates are used by the application to calculate a design using a Voronoi diagram, which is automatically clipped to the dimensions of the lamp shade. As the form of the lamp is altered, the shades and patterns are adjusted to fit. Once the code is compiled, a graphic preview is displayed. For the pilot version, users could use key-commands to toggle between a view of the form of the 3D form of lamp, the Voronoi-diagram pattern, and a 2D preview of the press fit parts (fig:5-4.) A final key-command allowed for the resultant design files to be exported as three separate PDFs, containing the paths for the press-fit frame, the shades, and the pattern files.

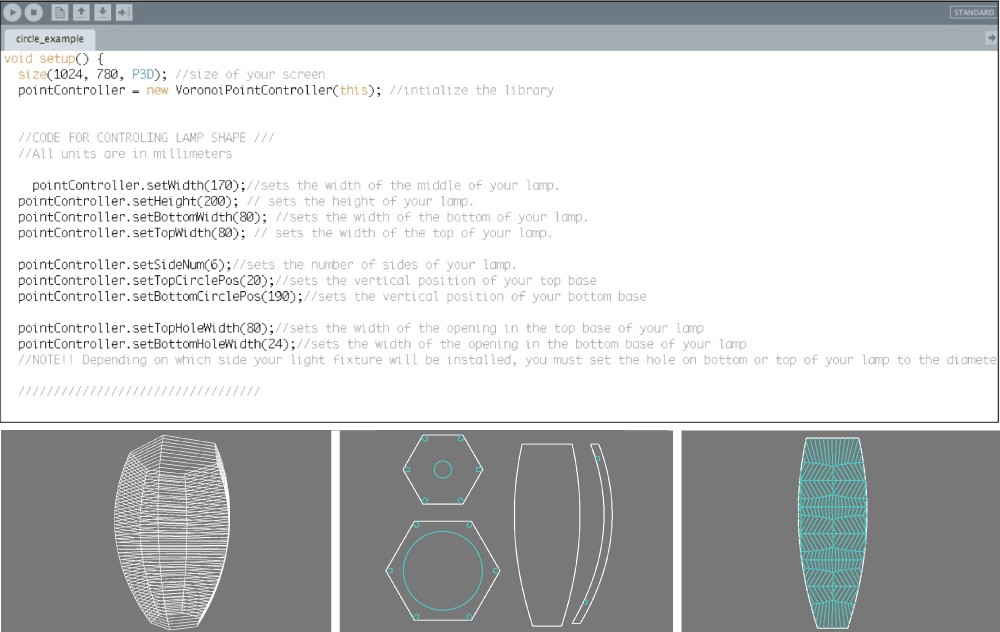


Figure 5-4: The first version of Codeable Objects, with only text-based interaction

The functionality of Codeable Objects was designed to provide platform that allowed for flexible computational design and the creation of complex forms and patterns, while greatly simplifying the process of translating the design to a suitable format for digital fabrication. The laser cutting process produced parts that expedited the construction process, but still but required care and dedication in assembly. Combined, these properties were designed to foster an experience that merged programming, digital fabrication and hand-craft and (hopefully) resulted in a useful finished artifact.

**5.3 Evaluation**

The first evaluation of Codeable Objects was conducted with a group of nine graduate students, ranging in age from 24-34, during a six-hour workshop. Five participants were women. According to self-reported pre-survey data, all but one of the participants were intermediate to experienced programmers. Five of the nine had previous experience with Processing. The participants indicated they had little or no prior experience in design. What experience they had was primarily gained in high school art classes and college elective courses. During the workshop, each participant engaged in the design and fabrication of their own lamp. Participants received instruction in the use of Codeable Objects and a basic explanation of the principles behind the geometry of the lamp. The pilot version of Codeable Objects was packaged with a set of example programs that contained the

basic code for initializing the library and defining the parameters of the lamp, along with a variety of point generation methods. The examples included algorithms to generate spirals, circles and grids in points. Participants were also provided with access to construction materials, and received training in the use of the laser cutter. They given approximately four hours to design the structure and

ornamentation of their lamp, followed by two hours for fabrication and assembly.



Figure 5-5: Several of the finished lamps from the first workshop

**5.4 Results**

All but one of the participants in the lamp workshop successfully completed their lamp. The one exception was a user who wished to incorporate a specialized light fixture into their piece, but unfortunately damaged her parts while waiting for the fixture to arrive. The participants with little or no prior programming experience primarily relied upon tweaking or remixing the example programs to design the form and pattern of their lamp. Those with more programming experience experimented the library to produce a wide range of forms and patterns. One participant wrote a program that decomposed a black and white image into a point cloud and used that as the basis for her pattern. Another participant wrote a program that used a Gaussian distribution of points to achieve the gradual variation he desired in his final pattern.

The physical assembly process required additional time beyond the duration of the workshop for most participants. This was partially the result of the bottleneck on the laser cutter, however the craft and construction components of the project took longer than expected. Despite this, all the participants returned after the workshop to complete their projects. Some participants chose to add additional steps to the construction process, such as sanding their parts. The physical objects produced were both attractive and functional; participants displayed their lamps in their offices and

homes after completion. One participant returned several days later to build a second lamp so that he would have a matching set for his bedside tables (figure:5-5.)

**5.5 Discussion**

Because of their prior expertise, the experiences of the majority of the participants in the first study are not indicative of the feasibility of Codeable Objects for novice programmers. Their experiences instead stand in contrast to the experience of the novice coders in the successive workshops and provide important information about the usability and workflow of the software. Some of the expe- rienced programmers provided immediate practical assistance by developing new example programs for Codeable Objects, including an extremely popular program for cosine and sine wave pattern creation. Because the library was open-source, they were also able to submit upgrades to the inter- face design and functionality during the workshop itself, which were later incorporated into the core version.

The experienced programmers in the lamp workshop exhibited limited knowledge of computa- tional design prior to the start of the workshop. When asked in the pre-workshop surveys how they thought programming, design and craft could be combined, people were either uncertain, or described the combination as method to create dynamic interactivity, rather than a tool for the design of form and pattern:

*“You can combine software and hardware and make craft more dynamic (e.g. sen- sors).”*

Lamp Participant pre 1

*“[programming] gives [you] the ability to make something dynamic.”*

Lamp Participant pre 3

Following the workshop, participants demonstrated an awareness of some of the specific affor- dances of computational design, and the benefits of combining it with digital fabrication:

*“I understand now how programming can be used for quick prototyping and mockups that can be used to inform final design decisions. This is easy [and] helpful when using physical materials where mistakes can be costly.”*

Lamp participant 2

*“Using programming in the design process adds some exciting and unique capabilities over traditional design and crafting, including mixing in different algorithm and ideas from other existing software, and rapid prototyping of complex designs.”*

Lamp participant 6

Participants were also pleased with the creative affordances of the tool, and described how the software enabled them to expand their programming abilities to the realm of art and craft with greater success:

*“I think programming makes designing more accessible because you don’t have to be able to draw or paint.”*

Lamp participant 4

*“I love the idea of being able to combine my interest in programming for creative expressions.”*

Lamp participant 6

One participant remarked that she had always believed that she was terrible at art, but that making the lamp had altered that perception. Although the primary objective behind Codeable Objects was to make a form of algorithmic craft that was accessible to non-programmers, in the first workshop, it provided a pathway for programmers to apply their skills to design and craft. From these responses, it is apparent that even among experienced programmers, algorithmic craft has the potential to expand people’s understanding of the applications of programming and motivate them to apply computation to other forms of production and expression.

Several participants also put forth detailed critiques of the programming process, which brought up larger questions on the practice of computational design. One participant reacted against defining the generative qualities of the Voronoi diagram patterns as a design method:

*“Changing the parameters didn’t always generate the pattern you have in mind. It was more like generating a few semi-random patterns and you choose one that looks good. It is rather a trying-and-choosing rather than designing /making something you planned to have. I think ”design” involves ”intention” and ”planning.” Programming, crafting, and design should be combined in the way that entails prior planning and intentions as opposed to cutting together the semi-random choices, which could be good but I wouldn’t call that design.”*

Lamp participant 6

As this quote indicates, the attributes of randomness and generativity do not automatically lead to optimal or good design decisions. Some deciding factor has to play a role in the process, but the criteria for this decision are often ambiguous. This criticism touches on a core debate about the role of conscious design and the restriction of intuitive creativity in computational practices. The emergence of comments like this are encouraging, because they reflect the engagement of the participants, not just with the task at hand, but in a critical evaluation of the creative implications of this form of creation.

**5.5.1 Challenges**

There were also elements of the process that were problematic for the participants. It became clear during the workshop that textual programming was not the best method to design the form of the lamp. Many of the participants became frustrated about having to set the parameters and then wait for the compilation process to complete before they could view the resulting form. This issue was addressed partly in subsequent versions of the tool by replacing the textual parameters with a set of sliders in the compiled application, which would adjust the form in real time, across each of the views (figure: 5-6.)

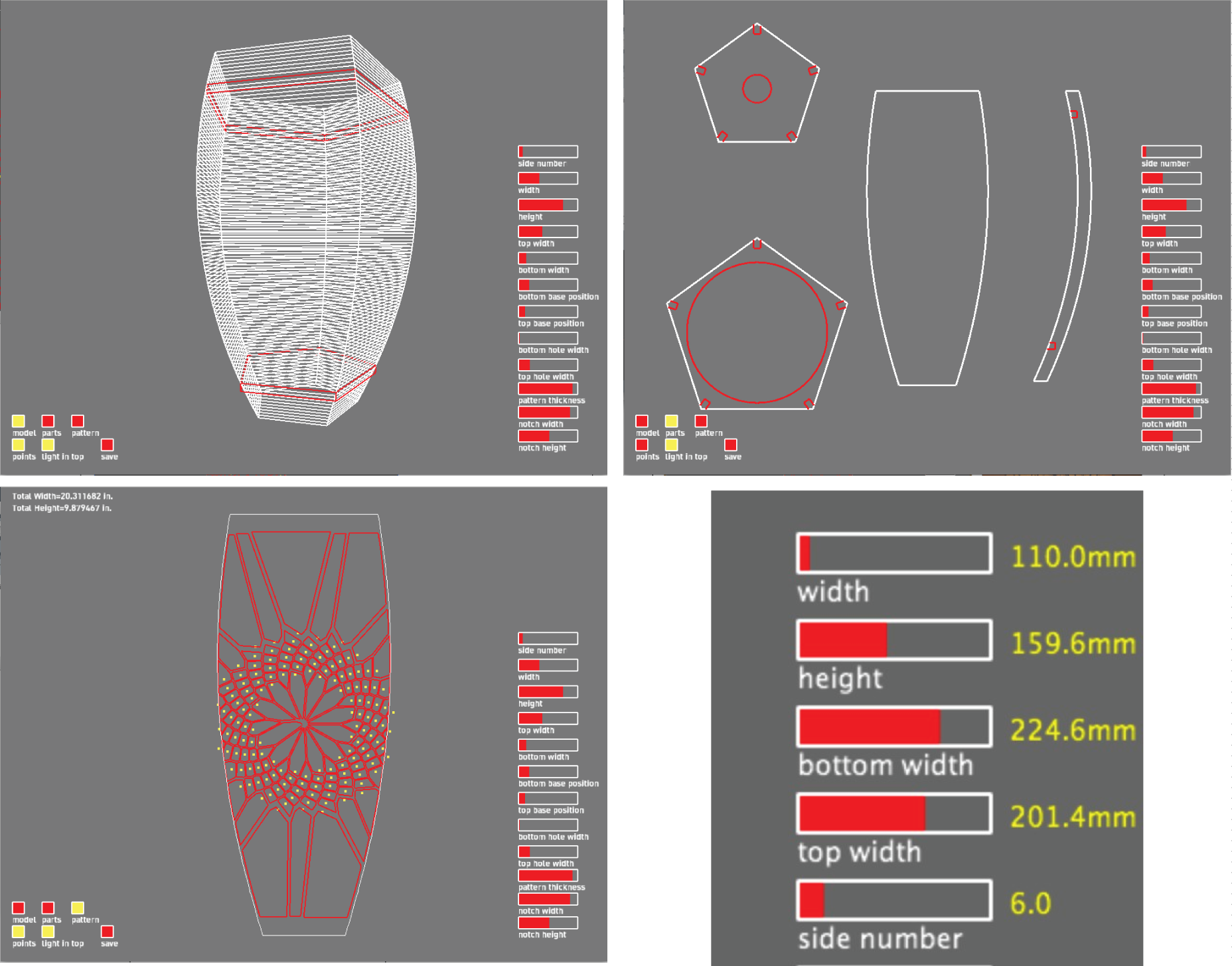


Figure 5-6: Revised graphic view with sliders

The textual programming method was useful for creating pattern on the shades. The simple method of specifying points in a programming context provided a space for creativity and resulted in greater variation in the lamps. If the tool had relied on a more standard set of graphical user interface(UI) components, like sliders to control the point generation, it is doubtful that the same range could have been achieved. On the other hand, it was clear that the less experienced program-

mers had more difficulty deliberately designing the patterns of their lamps, and relied primarily on adjusting and remixing existing examples.

Another restriction of Codeable Objects was apparent in the design process. Although adjusting the parameters and input values to a system can be considered a form of design, it only touches the surface of the design opportunities made possible algorithmic craft. With Codeable Objects, people cannot modify the program which defines the range of forms and patterns that are possible, unless they alter the source code of the library. This limitation prevented the evaluation of some of the more interesting computational design processes such as allowing people to create personal stylistic abstractions for patterns and forms, by writing their own algorithms. The stylistic limitations of Codeable Objects contributed to the high success rate in project completion, and the general attractiveness of the resulting projects, but limited the design flexibility.

One other defining component of the Codeable Objects pilot workshop was the contrast between the difficulties that arose through computational design and digital fabrication and the challenges in crafting. The difficulties people experienced while designing and fabricating their projects were often discrete; such as correcting for mathematical error in coordinate placement, or having the incorrect setting on the laser cutter. More complex problems arose in these contexts as well, such as confusion around principles geometric point placement, however they were problems that could be addressed through verbal instruction and explanation. The challenges encountered in the crafting session were of a material or physical quality. People struggled with finding the best techniques for assembling the parts and finishing individual pieces so that the resulting product maintained an attractive appearance. Most participants were surprised at the amount of time required to complete the physical assembly, and were sometimes frustrated when variations in the crafting process violated the precision and perfection of the digital design, and laser cut parts.

Some of the frustrations in the physical construction process were addressed in subsequent work- shops by creating a paper variation of the lamp that was faster and easier to assemble and required no gluing (figure: 5-7.) In addition, a feature was added to the software which reported the approx- imate material size required for a design, so that users could ensure theirs would fit on the bed of

the laser cutter.

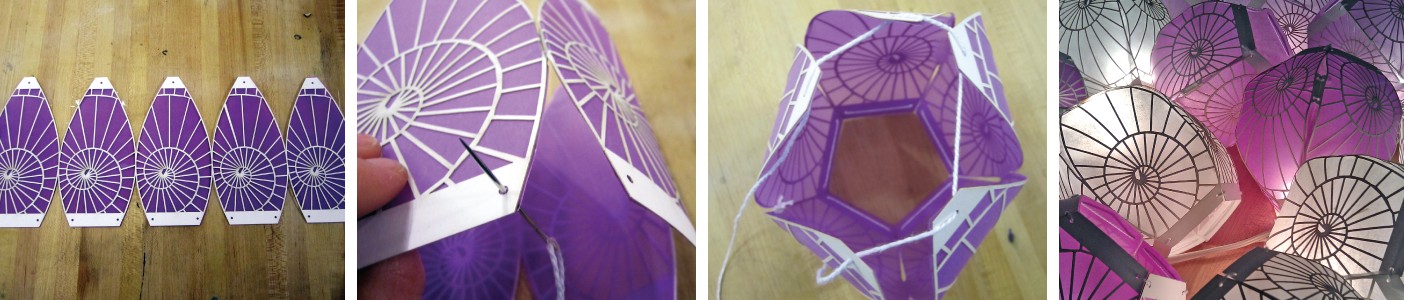


Figure 5-7: The revised paper lamps

Although there is often an opportunity to improve a user’s experience through changes in the interface and artifact design, it is useful to distinguish between frustrations that are the result of a software interface, and learning processes that are inherent in design and craft. In both craft and computation, care and practice contribute to more successful results. Furthermore in craft, material variation and inconsistency can be difficult to manage, but it can also contribute to the uniqueness and value of an object. The process of supporting people in algorithmic craft, is not just about building tools that make the programing process easier and the crafting component less prone to risk. Instead, computational tools should be designed to remove technical barriers in programing and fabrication so that people can address the more interesting (and often more difficult) design challenges of envisioning and refining computational systems. Craft processes should be feasible to achieve, but also recognize that part of the pleasure and accomplishment of crafting comes from the risk and difficulty it entails.

**5.6 Summary**

Codeable Objects was successful in that it allowed people to produce useful and beautiful objects with personal value, through a relatively pleasurable experience in computational design, digital fabrication and craft. It also produced in experienced programmers, a recognition of the aesthetic and design potential of computation, and brought up deeper questions about the role of conscious choice in computational design. The workshop demonstrated the importance of immediate visual feedback for informing design decisions, and also indicated the importance of applying textual programing to formats that clearly demonstrated its advantages. The workshop also highlighted the utility of open-sourcing algorithmic crafting tools, as this lead to immediate innovations in the functionality of the software when testing with advanced users. For subsequent tools, my goal was to build on these success while attempting to diminish some of the design limitations of Codeable Objects. I also began working with a representative group of designers: young people with no prior programming experience.

**Chapter 6**

**Soft Ob jects**



After an evaluation of the successes and limitations of the Codeable Objects library I made an effort to modify the library in a way that would allow for a broader range of computational design approaches and end products. In particular, I was interested in exploring the domain of algorithmically-crafted garments and fashion accessories. Fashion is an exciting domain to connect to computation, because it appeals to groups of people who are often under-represented in computer science, particularly women and girls. Computational fashion design has the potential to resonates well with algorithmic craft, because it offers the opportunity to apply digital fabrication to textiles

and fabrics, and introduces sewing and pattern-making as components in the construction process. Because garments and fashion accessories are created to be worn, computational fashion design requires the creator to consider questions of comfort, sizing and personal taste and and style when writing code.To support computational fashion design in the context of algorithm craft, I expanded Codeable Objects into a more general programming library entitled Soft Objects and evaluated it over a 10-day workshop with young designers.

**6.1 Motivation**

With the growth in public awareness of digital fabrication, there is a great deal of enthusiasm for fashion applications of digital fabrication technology. Much of this excitement is directed towards

3D-printed wearables and textiles. In July 2010, Iris Van Herpen released her Crystallization collec- tion, which featured her first computationally designed, 3D printed piece, marking the first time a

3D printed garment had appeared on the runway [4]. Van Herpen and many other fashion designers have continued using 3D printing as a medium for fashion since then. As a result, computationally designed, digitally fabricated fashion is often synonymous with 3D printing. The 3D printed gar- ments and accessories produced by professional designers like Van Herpen serve as inspiration for the future of digital fabrication, and demonstrate wearable forms that would be impossible to create through any other means. For the average person however, computationally designed and digitally fabricated garments of this nature present considerable limitations. Given current technology, cost and material restrictions, the majority of 3D printed garments are impractical for every-day wear and require advanced fabrication techniques that are unavailable to many non-professional designers. Garments like the N.12 bikini, designed by Continuum [17] are intended to be more practical, and available to consumers, but for the time being they are limited in scale, and still come at a steep price point. The construction of 3D printed garments of this form appear to have little in common with existing methods of garment production, like sewing, knitting and embroidery. Though garments of this nature are inspirational and groundbreaking, individuals with sewing, knitting or other textile manipulation skills may perceive their interests to be incompatible with computational design and digital fabrication in this form.

Although perhaps less publicized than 3D printed fashion, other other designers are merging fash- ion with computation and digital fabrication in a way that blends new technology with established approaches. Diana Eng’s Laser Lace tee collection contains laser-cut machine-washable t-shirts with floral-inspired iconography, and her Fibonacci scarf is created through traditional knitting tech- niques, meshed with a Fibonacci knit pattern. Eunsuk Hur’s modular fashion pieces are inspired



Figure 6-1: 3D printed fashion (from left to right: Crystallization 3D Top by Iris Van Herpen, Drape Dress by Janne Kyttanen, N12 Bikini by Continuum Fashion, Strvct shoe by Continuum Fashion

by tessellations and fractal geometry. By creating garments through laser-cut interlocking pieces, Hur’s aim was to produce items that were robust and durable, also gave the user the opportunity to use their inner creativity to come up with new and interesting items, by rearranging the individual components (figure:**??**.)

Examples such as these demonstrate a space in computational fashion design that is compatible with non-digital interests, and skill-sets. Subtractive fabrication machines, like laser cutters and vinyl cutters are dominant in this type of work, because they work with traditional materials and can produce garments at a much lower cost and larger scale than 3D printers. Variation in materials also can translate to a wider set of possibilities for hand crafting as well as different aesthetics and styles. It should be noted that accessible forms of 3D printing can produce compelling wearable objects, however they are generally on the scale of jewelry and small accessories. Garments designed and produced as a blend of digital fabrication and textile materials and construction processes correspond well with the values and practices of algorithmic craft. The Soft Objects programming library was designed to allow new programmers to design the forms and aesthetic components of fashion accessories and garments through programming, and then physically construct the garments using subtractive forms of digital fabrication and sewing.

**6.2 Tool description**

The Soft Objects library contains a set of methods that allows users to draw shapes and patterns and then export those shapes and patterns in a vector-file format that is compatible with x-y axis digital-fabrication machines. Similar to Codeable Objects, Soft Objects is used within the Process- ing programming environment and contains a set of programing methods that enable the design of visual forms and patterns. SoftObjects allows users to define and manipulate basic geometric primitives such as Points, Lines, Curves and Polygons. These primitives can then be collected

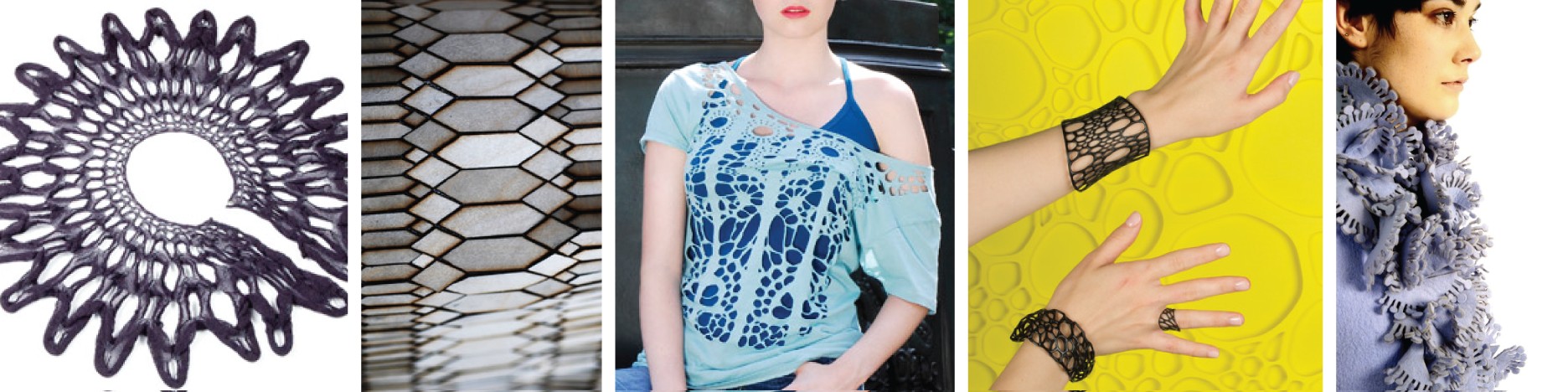


Figure 6-2: ”Ready to wear” computational fashion (from left to right: Fibonacci Scarf by Diana Eng, Biomimicry laser-cut bracelet by Stefanie Nieuwenhuyse, Laser Lace All-Over Tee by Diana Eng, Interstice bracelet by Nervous Systems, Modular Fashion by Eunsuk Hur

within Pattern and Shape objects—structures designed to capture surface decoration and 2D struc- ture, respectively—to form increasingly complex designs. Soft Objects is formulated on an Object Oriented Programming (OOP) paradigm, which lets users create and manipulate collections of ge- ometric primitives—Patterns and Shapes. This structure differs from Processing’s drawing API, which uses a functional programming approach. Users are presented with a 2D preview of their designs when they compile their code. The structure of Soft Objects enables users to simultaneously apply transformations to all of the elements in a collection that make up a complex pattern or shape. The objective behind this structure was to open the design possibilities in a format that was suitable for creating complex 2D designs and aesthetic patterns for clothing and accessories, while ensuring that a user’s designs would be compatible with subtractive fabrication. To simplify the process of garment creation, Soft Objects included functionality to import existing cut patterns as scalar vector graphics files (SVGs). This allowed users to merge programmatically generated designs with pre-sized shirt, dress and pants pattern-templates.

Soft Objects supports a variety of digital-fabrication machines by allowing users to save designs to vector portable document format (PDF) files. PDFs can be used by different production tools, including ink-jet printers, vinyl cutters, laser cutters, and computationally controlled embroidery machines. Output from Soft Objects can be fabricated on essentially any x-y axis tool. 3D structures can be created by assembling fabricated pieces. Figure 6-4 demonstrates the workflow from code to a finished object. The Soft Objects library also contains a collection of pre-defined algorithmic patterns that can be initialized, including Voronoi diagrams, Koch curves, and L-Systems, and an extensive set of example programs that users can modify and combine to produce individual results. These examples and pre-defined algorithms were created as a way of quickly exposing new programmers to some of the complexity and variability that is possible through computational design, and demonstrate the abstract qualities of computation, without immediately requiring them

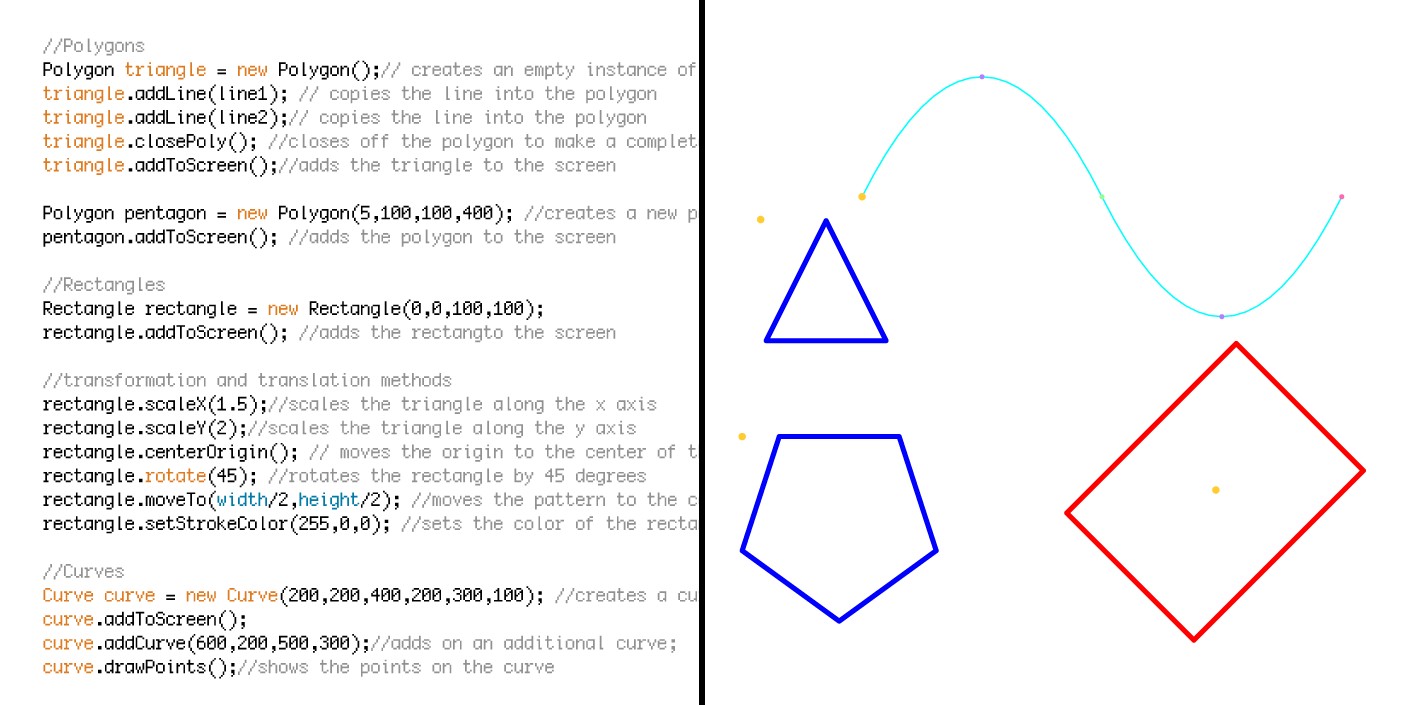


Figure 6-3: Soft Objects primitives

to learn substantial amounts of programing structure and syntax.

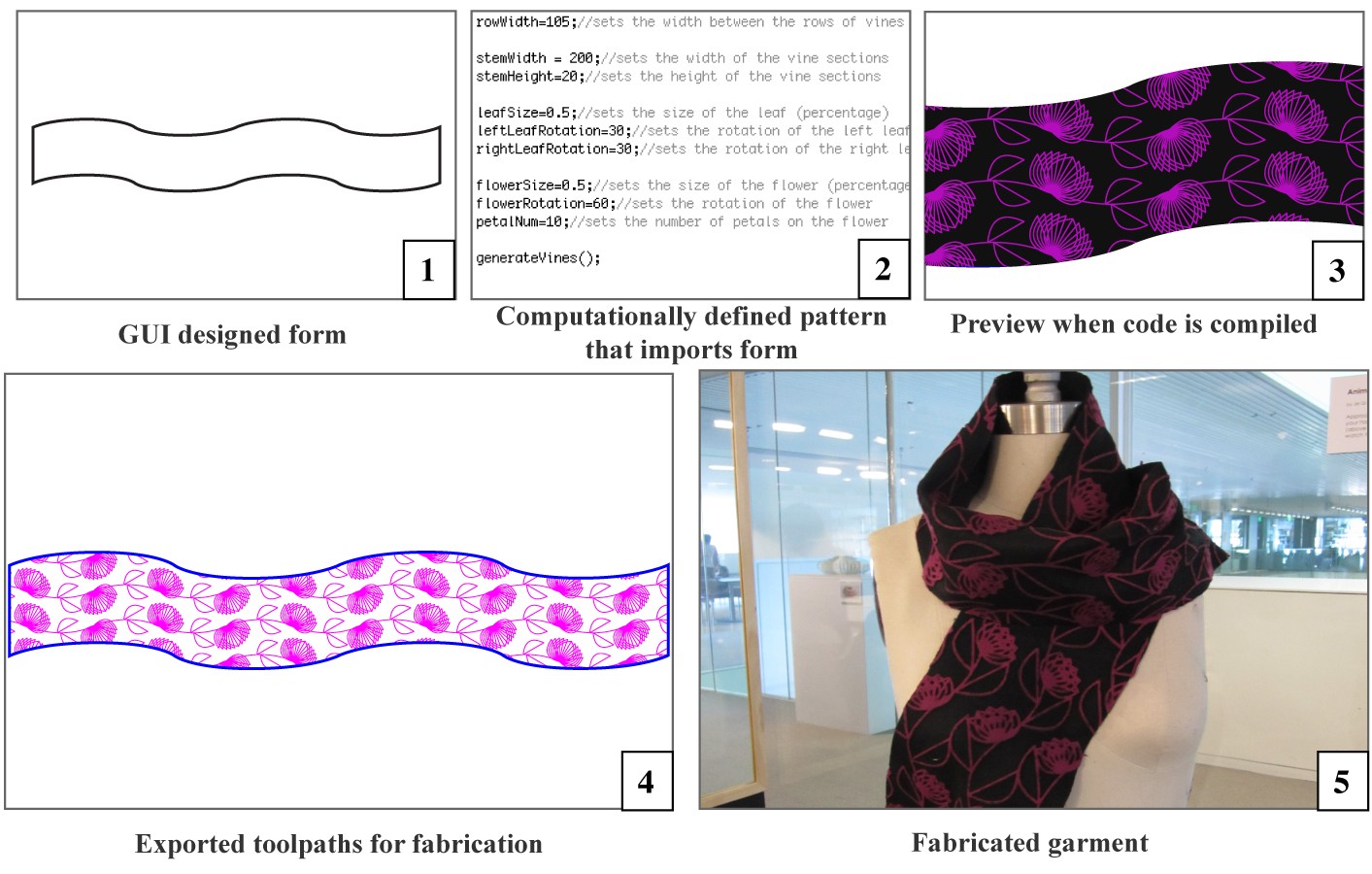


Figure 6-4: Soft Objects workflow

**6.3 Workshop**

The evaluation of Soft Objects was conducted during a 10-day workshop with a representative group of participants—eight young adults, aged 11-17, 75% male and 25% female. A significant majority (88%) stated in pre-surveys that they had little or no prior experience in programming, and only one participant had prior experience in Processing. All of the participants indicated some level of prior experience in art, design, or craft. Most attributed their design or craft experience to art or drawing classes. The workshop was conducted at the Nuvu Magnet Innovation Center for Young Minds. Participants were given 10 days to conceptualize and construct a garment using a combination of computational design, digital fabrication, and sewing and crafting. The workshop was more open- ended than the Codeable Objects workshop; participants could produce any type of garment they wished as long as components of it were computationally designed and digitally fabricated. During the workshop, participants were introduced to Soft Objects and the concept of computational fashion through a multi-step process that engaged participants in different levels of programming through the construction of different garments and accessories. First, participants were provided with a small set of example programs similar to the lamp workshop. This step allowed them to manipulate a core set of parameters to generate the pattern and form of a scarf, which they then cut on the laser cutter (figure: 6-5.)



Figure 6-5: Bracelets and scarves from preliminary activities

Second, participants were instructed in a number of primary programming concepts, including iteration, function definition, and the use of variables and primitive data-types. During this in- struction, participants were guided through the process of independently using Soft Objects and generating their own programs from scratch. They used these programs to create a design for a wooden bracelet (figure: 6-5), which was then laser cut and assembled. These two initiation activ- ities were intended to demonstrate the stylistic affordances and design properties of computational design and digital fabrication, as well as providing participants with the practical foundation to

begin conceiving of and executing their own ideas. During the remainder of the workshop, the par- ticipants were asked to conceive their of own garment concepts and designs and provided with the resources to design, prototype, and craft finished artifacts.

**6.4 Results**

Participants in the fashion workshop were successful in using programming and digital fabrication to design and produce finished garments. During the initiation activities, participants independently wrote and compiled programs of their own and produced physical products based on the design gen- erated from that program. Furthermore, with assistance from the instructors, the participants were able to apply more sophisticated programming methods to produce a diverse set of final products (figure: 6-6). One pair of students developed an “armor dress by writing a program that geomet- rically described a single “scale shape, imported a dress pattern from Illustrator and filled it with rows of scales that corresponded with the dimensions of the dress. Another pair created a geomet- rically inspired dress with a patterning of different-sized octagons and squares that were laser cut from starched fabric. Another student created American-flag-inspired pants using a program that generated random orderings of red and blue stripes on a white background. One group that was less interested in the process of sewing clothing created a program that generated a recursive virus-like pattern and then screen-printed the pattern on pre-made sweatshirts and t-shirts.



Figure 6-6: Completed garments (from left to right: octagon dress, flag pants, samurai dress, viral sweatshirt)

On the post survey, when asked if they “were able to complete a finished project to their satis- faction, 100% of the fashion participants responded in the affirmative. The resultant garments were attractive and functional, indicated by the fact that participants from the fashion workshop kept and wore their creations. Direct comparison of the pre-and post-workshop surveys also demonstrated

that on average, participants in the workshop indicated their interest in crafting increased after the workshop, as did their enjoyment of the design process. 88%of the participants in the fashion workshop indicated that they felt more comfortable programming after the workshop than before.

**6.5 Discussion**

The Soft Objects workshop occurred over a longer time period than the Codeable Objects workshop, and explored a wider range of crafting and computational techniques. As a result, I had an oppor- tunity to spend greater amounts of time observing and talking with the participants. Because the participants were novice programers, their experiences better reflect the target demographic of this research. Through this workshop we confirmed that algorithmic craft activities can actively support the expression of personal identity among young designers. It also fostered feelings of confidence in programming and supported aesthetic and technological literacy. The workshop promoted a deep understanding of computation as evidenced by critiques of the participants as well as demonstrating the importance of physical prototypes in the design process. Finally, the workshop promoted a sustained engagement in programming.

**6.5.1 Identification as a programmer**

Similar to the participants in the Codeable Objects workshop, the fashion participants began with vague ideas about the applications of computation. When asked in the pre-workshop surveys how they thought programming, art and craft could be combined, participants responded in writing in the following ways:

*“To be honest, I am not sure too sure how it would all be combined because I don’t know much about programming.”*

Fashion Participant Y

*“With programming, we can make programs do things for us.”*

Fashion Participant J

In addition, the participants had almost no prior knowledge of computational design. Those par- ticipants who had some form of prior programming experience associated it largely with interactivity and actuation:

*“I would love to combine things like T-shirts and speakers or other different types of technologies.”*

Fashion Participant J

Following the completion of their final projects, participants were much more descriptive about the applications of programming:

*“I think [programming and fashion] are really interesting but I never thought they could ever be together in one concept, and it’s awesome that I know that now- that you can design aesthetically pleasing things from coding.”*

Fashion participant K

*“I’ve never thought of programming as physical, I thought it was only in computers. But then when we made the scarves and stuff, I thought that was really fun.”*

Fashion participant M

Many people also expressed a growing confidence in their ability as programmers. During the interviews, several separately stated that that while they did not feel completely comfortable pro- gramming independently, the experience made programming feel significantly more accessible:

*“For someone who never had any programming explained to them before, when you look at [computational drawing examples] it feels really inaccessible, but now that I’ve been taught a little bit of [programming], I can kind of crack at the walls a little bit and understand how it works, and that makes it more accessible to me.”*

Fashion participant S

*“Even though now I’m not really a Processing expert now, I’ve just experienced it and it’s not as scary to me, like the idea of coding, you just kind of have to learn some stuff and practice it more, but I think I definitely understand the concept of it.”*

Fashion participant K

Finally, the programmers in the fashion workshop expressed feelings of pride and a sense of accom- plishment in their new-found programming skills:

*“I’m actually kind of proud. I know what’s going on. It feels different... I just thought [programming] was just about these huge programs that you have to piece together, and that you have to be really, really smart to do it, but I can do it.”*

Fashion participant P

*“I think it was really cool that we used [programming] for fashion, cause I think a lot of people might think people who do fashion aren’t really smart or something, and then they think that people who design code are like brilliant coders and can do really awesome stuff with it.”*

Fashion participant K

The confidence, sense of belonging, and personal agency demonstrated in these comments stand in contrast to popular views of programming as specialized and inaccessible. These sentiments indicate that introducing programming in the context of algorithmic crafting not only has the potential to change people’s understanding of the relevance and applications of computation, but also promote a personal awareness of technological literacy and competence.

**6.5.2 Application of computational affordances**

Aside from a general understanding of the potential applications of computation, participants were successfully able to leverage the advantages of computational design in their final projects. For ex- ample, the octagon dress used parametric design principles as the participants were able to change the size and orientation of the octagons in the dress by modifying several parameters at the start of their program to affect the entire pattern, rather than rotating and adjusting each shape in- dependently. The samurai dress took advantage of the computational properties of precision and automation. With help from an instructor, the creators programmatically generated an individ- ual vector file for each row of scales of the dress, expediting the production process (figure:6-7.) The samurai dress is particularly interesting because it shows a remarkably successful transition from the design expressed in original concept art by the participant, and the resultant garment, demonstrating the adherence to the person’s creative vision through computation. Lastly, the viral shirt, demonstrated an application of generativity. The aesthetic of the viral pattern was produced through the use of a weighted random-number generator to determine the number and length of the branches in each recursion of the pattern. Although the implementation of the weighted number generator was facilitated through the help of the instructor, the participants came up with the idea of using it on their own.



Figure 6-7: Progression of samurai dress (from left to right: concept sketch, computationally gener- ated pattern,laser-cut components of final garment)

Many of the participants demonstrated an understanding of the rationale behind the methods they were using. One of the creators of the armor dress project compared the process of programming the design of the dress to that of manually drawing it:

*“With drawing you can achieve everything programming can, but I would prefer to program it. [programming] can be pretty convenient...the computer is helping me. Like if you want to make pizza, the computer is like a pre-made crust.”* [Fashion participant E]

Participants also articulated an understanding of specific programming functionality. One par- ticipant described the point at which she understood the application of parameterization:

*“One moment that stuck out was when you helped me make a code with original geometry that could be changed so that when you changed one thing it changed everything and that was cool because I felt like I actually made something that could be changed and then applied.”* [Fashion participant

K]

The ability to understand and describe how computational support one’s creative objectives is essential in motivating an individual to spend time learning and implementing these methods. Once the aesthetic possibilities of the recursive viral pattern were apparent to the designers, they became engaged in better understanding the underlying algorithm, so that they could produce a pattern to their exact specifications. Algorithmic craft can provide the motivation for participants to tackle complex computational problems with sophisticated approaches, when problems are clearly grounded within the design objectives of the individual. A continual challenge in engaging new programmers in algorithmic craft is in selecting programming approaches that allow for compelling aesthetic and design possibilities, but remain approachable for first-time coders.

**6.5.3 Aesthetics and Identity Expressed through Code**

The fashion workshop provided the opportunity for participants to use computational aesthetics as a way to express their personal style. Fashion can serve as a means of self-expression and for conveying one’s identity. Discussions on fashion conducted at the start of the workshop indicated that participants were aware of the connection between fashion and identity and were eager for opportunities to create clothes that expressed their style. As a result, the majority of the garments created in the workshop contained an expression of the “fashion sense of the participants who created them. For example, the participant who created the flag pants was very explicit that the pants have some form of an American flag motif, but not resemble the traditional, and as he put it “tacky flag pants that he commonly saw. He wanted his flag pants to be “something that he would actually want to wear. His programming choices were made in direct consideration of his desire to create a pair of pants that he felt were fashionable.

When asked about the experience of making and designing his pants he said:

*“[The workshop] definitely changed my impression of making clothes, I thought it was pretty quick to make clothes, but it actually takes a long time, and it’s also really fun. I love the fabric I made.”* [Fashion participant M]

His enthusiasm also was evident in the fact that after the pants were complete, he tried them

on and wore them for the remainder of the workshop. This level of enthusiasm was common among participants; they all proudly modeled their creations, and many of them wore them home. This behavior suggests a relationship between the decisions made in a programming context, and the participant’s desires to express their visual identity. The participants were selective in the code they wrote to design their garments because they intended to wear the garments, and as a result, be represented by them. This powerful affective relationship between computation, design and self- expression provides a natural way to engage people in programming and design by supporting their personal interests.

**6.5.4 Physical and Digital connections**

One of the challenges of Algorithmic Craft, alluded to in the Codeable Objects discussion, is that the practitioner must work between digital designs with physical materials and processes. Physical prototypes often serve as a key point of transition between these spaces. In the fashion workshop, prototyping played an important role, and demonstrated how computational tools can support and sometimes hinder the prototyping process. The focus on fashion made it possible to supply the participants with large amounts of inexpensive test fabric. The laser cutter could cut fabric much more quickly than thick materials, which allowed participants to produce numerous prototypes of their projects before creating a final piece. Most groups produced two or three prototypes, with one participant creating six iterations of a single jacket. This rapid production process formed a direct connection between discoveries made in the physical prototyping space and decisions in the programming realm. In the case of the octagon dress, (figure:6-6), the participants first cut test rows of octagons to determine the appropriate scale, then adjusted their design by modifying their program. When they had cut out a second more complete version of the dress, they rotated one of the shoulder straps on the physical prototype and formed an idea for a one-sided shoulder strap. They implemented this design change in the digital version of the dress by making additional changes to the size and rotation of the shapes defined in the code. When asked about this process in the interview one of the participants said:

*“I think it was really fun that we got to do a prototype first because then if you don’t like it, you don’t feel a lot of pressure because you can make it again really fast, and there’s no stress because if it doesn’t turn out well, then it’s not your final project.* [Fashion Participant K]

The combination of programming, rapid fabrication, and physical construction allowed for a design approach that transitioned from programming to fabrication to programming adjustments based on the fabricated elements, and then back to fabrication. This iterative approach resulted in a closely linked cycle of physical and digital engagement.

**6.5.5 Enthusiasm in crafting and coding**

One of the most encouraging aspects of following the fashion workshop, was the participant’s en- thusiasm and desire to continue making. Participants talked extensively about what they would like to make in future with programming, consisting citing that they would like to continue making clothing, or other personal functional items like furniture and “things they could use around the house. The experience of both sets of workshop participants also demonstrated the ability of these techniques to produce objects that were designed to complement personal items and living spaces. When asked what she would like to make if she continued to program, one participant responded:

*“Things like we’re making now, things that you would want to keep or use, things that look nice as opposed to like computer games, or “input-output devices. I think those are fun, but it’s not as cool as things that you can hold in your hand. I actually hung up the scarf I made in my room, and now I can be like “I made this on Processing and people will be like what? It’s cool! ”* [Fashion participant K]

This enthusiasm, combined with the high potential for individual expression and sense of ac- complishment encouraged us to continue exploring fashion and garment production as topic space for algorithmic craft. While the fashion workshop highlighted many positives in this sense, we also encountered several areas for improvement.

**6.6 Limitations**

The most evident barriers in the Fashion workshop involved the syntactic challenges of programming. Many participants expressed a frustration with the syntax in both surveys and in-person interviews. Although the workshop participants were able to generate their own programs, they required more assistance from an instructor to write some of the commands. In addition, a feeling of needing to memorize programming syntax frequently translated to a sense of frustration. One participant stated in an interview:

*“I couldn’t memorize things, so it also was frustrating for me to always have to get you to help me write the code.”* [Fashion participant K]

Many people requested some form of written ”cheat sheet” that listed the key methods and how to use them. They also pointed out that you often had to write a lot of code (including import commands as well as setup and draw functions), even for simple tasks. Writing code for the first time is always challenging, however the high levels of frustration registered by the participants often focused on aspects of programming were extraneous to the design itself. Because I wrote Soft Objects as a Processing library, it required that the syntax correspond to Java, a difficult language for beginners. Java is a general application language and has many syntactic requirements that are unnecessary for computational design applications. Based on this difficulty I concluded that future

algorithmic crafting tools should explore domain-specific languages that directly applied to design and fabrication.

Along with difficulties with the programming syntax, participants struggled with some of the post-processing techniques. In order to be suitable for digital fabrication, many of the participant’s computationally-generated designs required some post processing in Adobe illustrator. Usually this required using illustrator’s shape boolean functionality to merge shapes or expand outlines so that the vector paths would correspond to the desired cut pattern on the laser cutter (figure:**??**.) Al- though the methods to perform these operations were simple, they needed to be repeated every time the design was modified programatically. This was not only inefficient, but sometimes pre- vented people from determining if their designs were feasible for fabrication when they were in the programming environment. This difficulty encouraged me to focus on ways of removing the need for illustrator from the process altogether so that all design modifications could be initialized and updated programatically.

Many participants also struggled with the concept of prototyping. These struggles were evident in practical aspects, such as participants not saving their programs and digital design files to come back to later (despite repeated reminders from the instructors to do so). Participants also frequently spent too much time on assembling their early prototypes and were frustrated when they realized they would have to make design changes and repeat some of the manual labor. Although we took pains in the workshop to introduce participants to the concept of prototyping, these instructions were not always absorbed. This may present an opportunity for future algorithmic crafting tools to contain specific features to encourage and support the physical prototyping process. One possibility is to include simulation tools that preview the constraints and behaviors of physical materials. The value of working with physical prototypes should also be supported however, by features that allow the designer to scale their files and fabricate first in miniature, and software that allows easy access and management of multiple versions of a design throughout the design process.

Finally, as in the Codeable Objects workshop, participants were frustrated by the delay between adjusting their code and seeing the results. Although, not an absolute solution to the challenges in learning programming syntax, a programming environment with more immediate feedback could assist with the issue of syntactic challenges by providing novice users with improved ways to visualize the effect their syntactic changes have on their design. The implementation of background compiling, the process by which code is automatically compiled and executed as changes are made, has been applied successfully in several tools for novice programmers, including Scratch and Alice [16], and more recently with Khan Academy [10]. Our reliance on the Processing for Soft Objects made the incorporation of real-time compilation infeasible, however it quickly became a goal for future tools.

**6.7 Summary**

Participants in the Soft Objects workshop were able to create personal wearable items that were beautiful, functional, and personally meaningful through computational design, digital fabrication and craft. During the workshop, the participants learned new skills in programing and digital fabrication, as well as techniques in pattern-making and sewing. The participants emerged with an awareness of applications of computational design for digital fabrication, specific understanding of some of the primary elements of programming, and a desire to continue using computation to build their own objects. Although there are many areas for improvement in future tools, the Soft Object workshop primarily demonstrated the importance of incorporating approaches that further reduce the practical challenges of programming. Possible approaches include removing extraneous syntax, reducing the number of programing methods, providing rapid and informative visual feedback, and adding features that do a better job of helping users make their designs feasible for fabrication. The objective of these features is not to trivialize the process of algorithmic craft, but to remove the barriers to independent and deliberate creation through computational design, and to ensure the successful translation of digital designs to physical forms that