Foil:
A Three-Dimensional Design Sketchbook

Master's Thesis Paper
New York University, Interactive Telecommunications Program
May 2011

Patrick Hebron

Introduction

Three-dimensional design tools are not for the faint of heart. In professional design environments such as Autodesk's Maya, menu items carry cryptic names like "Jiggle Disk Cache Attributes" and even the most basic functions require a three-button mouse or multiple-keystroke commands. I have used tools of this kind for more than a decade, but have never found myself able to utilize them in the formation of my ideas. I generally work a design concept out on paper through a series of interrelated diagrams and only turn to software when I am ready to execute a final model. This experience led me to the notion that most computer-aided design software is geared towards the execution of an idea rather than the development of one. There is a clear need for professional design tools such as Maya and AutoCAD, but even in this context, I believe that the creative processes of corporate teams and individuals alike would be well served by tools that are aimed at fostering an unrestrained process of idea generation. To address this idea, I began a yearlong process of developing a tool called Foil for my thesis work at New York University's Interactive Telecommunications Program (ITP). Foil is a three-dimensional design tool inspired by the freedoms of a paper notebook and enhanced by an arsenal of intelligent aids, which help the user to easily and elegantly develop rough conceptual sketches and turn these into a finished model. It is an environment that facilitates experimentation with and the development of ideas. Most importantly, it is a tool aimed at helping rather than hindering the creative work of novice and experienced designers.

Background & Research

The blank page is a tool of inestimable power - it is open-ended, a surface adaptable to all manners of thought and expression. In a sketch book, the artist or designer may tack on detail views, side notes and visual markers of his or her own invention, which help to illustrate the ideas under development. Despite its many virtues, this medium is limited in its ability to enable its user to reapply the same components in divergent variations. A digital tool, on the other hand, allows its user to save numerous versions of a project without harming the original, but tends to enforce a particular way of working, which is aligned to the strengths and weaknesses of the program's interface. One common weakness is that in trying to reduce clutter, many digital interfaces adopt a document-oriented layout, which makes it difficult to work back and forth between different component diagrams, sketches and versions. So while digital design tools in some respects facilitate a heightened level of experimentation, they also handicap it through the rigidity of their interfaces.

Today, there are approximately four and a half million CAD users worldwide. This is less than half the number of Photoshop users. In the last two decades, digital image manipulation has become a commonplace activity of the mainstream computing culture. Three-dimensional design, on the other hand, has largely remained a niche activity. This is in part due to the fact that three-dimensional design tools are difficult to learn. Due to the conceptual complexities of three-dimensional geometry and the ambiguities associated with its representation in two dimensions, it is easy to understand why the development of a full-featured 3D design platform can become a protracted endeavor. This, in turn, translates to a complicated, labor intensive process for the user. It is difficult to quickly sketch out ideas under these restraints and as a result, most 3D software environments focus on enabling the production of polished, final work for which a labor-intensive process seems justifiable. In the last few years, however, products like SecondLife and Google Maps have led to an an increased demand for low-polygon-count models for use as avatars and architectural stand-ins. Google SketchUp addresses this design market by offering its users a simplified set of modeling tools as well as access to an online network of prebuilt models and templates. Other recent tools such as ILoveSketch and Pixar's in-house set construction software, Eden, have begun to shorten the learning curve by offering three-dimensional modeling platforms that draw heavily upon the user-interaction concepts already in widespread use in the consumer markets of iPhone and iPad multi-touch applications. Yet, while these tools facilitate a more intuitive modeling process, they do little to merge the processes of idea generation and final refinement.

As rapid prototyping and 3D printing technologies become more inexpensive and widely available, we are likely to see an increase in the market for 3D modeling tools. Recently established online services such as shapeways.com and zazzle.com allow users to customize object and print designs, which can be manufactured in a variety of materials and mailed to the customer. Zazzle, which specializes in custom prints, has developed cloud-based image layout and manipulation tools in order to draw customers who do not own Photoshop or similar software. Shapeways, on the other hand, allows users to upload standard format CAD files, or, if the user does not own the necessary software, he or she can choose a template design from an online database and perform small customizations to its color and so forth. That is to say, Shapeways has not found a truly viable solution for their problem - they have not enabled their less tech-savvy users to truly express themselves in a three-dimensional medium. Therefore, there is a clear need for software that is capable of facilitating a wide range of three-dimensional expressiveness through tools that are both easy to learn and capable of supporting rapid processes of design iteration. This is where Foil comes in.

Design Software - A Place to Think

In developing Foil, I sought inspiration from my own sketchbooks, from my childhood and more recent years, as well as from the notebooks and manuscripts of some of my intellectual heros. This latter group included the technical notebooks of Leonardo da Vinci and Thomas Edison as well as the manuscript drafts of artists from various disciplines such as Beethoven and Allen Ginsberg. One of the most important themes I found in these works was the tendency to organize the space around a central diagram or version of the work and then use lines, arrows and other such notations to connect an aspect of this central version with an auxiliary space in which some key detail or divergent concept could be further explored in temporary isolation.

Looking at these works, I was inclined to see the design process not only as a means to an end, but as an art in its own right. Design is a conversation with one's self, with the perceived needs of an imagined user and with the laws of physics, logic and form. The final product emerges from this dialogue, but will only be successful in doing so if that dialogue has been a rich and well-balanced one. As Michelangelo said, "Every block of stone has a statue inside it and it is the task of the sculptor to discover it." This can rarely be achieved in a single pass. Instead, the artist or designer must often think back and forth between pieces, developing each component both in isolation and in consideration of the whole. It is this choreographic process of ushering the many pieces into delicate alignment with one another that makes the paper notebook such a powerful tool for thought - the page is endlessly adaptable to that process and to the recording of its narrative. I saw little of this in computer-based design tools and so I focused my ambition for Foil around the idea of developing a tool that would lend itself to this creative process while incorporating some of the capabilities that are unique to computer-based tools.

User Interface

Foil's user interface attempts to bring the best features of paper notebooks and computer-aided design environments into one easy-to-use software platform. It is meant to serve the user's own creative process rather than circumvent it. A Foil project has four main interface elements: text notes, two-dimensional diagram views, three-dimensional diagram views (also called "detail views") and the main project view. An instance of any of these interface elements can be placed or moved to anywhere on the screen. Each can also be associated with any piece of geometry in the project. In designing a bookshelf, the user might open a blank 2D diagram and sketch the profile of the shelf's sideboard. A "new version" button allows the user to clone the most recent version of the sketch and make revisions to it without affecting the other versions. After finding a suitable contour for the sideboard, the user might click the "Send to Project View" button and continue editing it there in the context of the other components of the design. Foil uses a complex data structure that tracks the various versions and view associations held by a particular geometry, making it easy for the user to move ideas through different regions and iterations of his or her project. In any of the views, the user may apply simple multi-touch gestures to translate, rotate and scale geometries. Text notes can be associated with any of these other entities to assist the user in narrating his or her creative process. To lessen the difficulty of orienting oneself within the screen representation of a three-dimensional environment, 3D diagrams or detail views allow the user to simultaneously take numerous vantage points on a model or some key detail and work back and forth between these varied perspectives. Basic 2D and 3D primitives can be easily generated with a button press. In the next version of Foil, I will introduce key modeling tools such as boolean operations, polygon splitting tools and the like. In general, the user interface is intended to offer a simple, intuitive platform for rough sketching as well as the fine-tuning of three-dimensional models. A key aspect of the task of extending this toolset while still maintaining ease of use is Foil's machine-learning toolset, which is covered in the following section.

I developed Foil in C++ for Mac OS X and the iPad. Due to the novelty of some of Foil's interface elements and because it is of great importance for me to understand every aspect of my software, I chose to write Foil entirely from scratch with no third-party libraries with the two exceptions of OpenGL and Cinder for multi-platform display windowing and rendering purposes only. This decision meant a great deal of additional work but allowed me the freedom necessary to design a user interface and data structures that suited my goals rather than limiting myself to the features available through existing libraries such as the Cocoa UI Kit. Additionally, this way of approaching the software development process during my thesis leaves me with an enduring toolkit of geometric and user-interface source code that I understand intimately and which can be applied to future projects in addition to the continued development of Foil.

Machine Assistance Tools

In recent years, computer scientists have begun to apply powerful machine-learning algorithms to computer-aided design processes. One of the most noteworthy examples of this is NASA Evolvable Systems Research Group's work on the use of genetic algorithms in the design optimization of radio frequency antennas. These efforts have demonstrated that genetic algorithms and other machine-learning techniques can be of great use to design challenges that primarily concern issues of pure physics or mathematics. Due to the nature of genetic algorithms and their reliance upon an objective and quantifiable metric for the evaluation of an individual design iteration, these tools have thus far found extremely limited applications in fields where such metrics are not readily accessible or possible.

The notion of incorporating these sorts of machine-learning tools into design software is an intriguing one because it suggests the advent of a period of "a-historical design." That is, when a person sits down to design a solution to a problem, he or she is in many ways tied to the historical legacy of how other people have already solved the component problems that make up the design challenge at hand. Genetic algorithms pose something of an alternative to this historical pathway because they are designed to care only about how well a particular iteration meets certain criteria. The approach a genetic algorithm takes to reach to a viable solution may have nothing to do with the premises we would be likely to evoke through our historically-tied thinking. While this possibility is an intriguing one, we must also see that in many areas of design, even if a genetic algorithm could be successfully applied, there is often a big difference between a great or efficient design and one that people would want to live with. Therefore, genetic algorithms cannot in themselves constitute a magic bullet for the advancement of the design history. In general, people are required to establish the significant questions and goals within a particular field of study and to mediate the divide between mathematical efficiency and human usability. Therefore, if machine-learning algorithms are to become truly useful to design, it will only be through a well-integrated partnership with human users.

Foil takes a unique perspective on the role of machine-learning algorithms in design software. Its purpose within Foil is twofold: it is designed to assist the user in keeping track of the various components, diagrams and versions of a design as well as to quickly generate variations of the user's designs in order to offer the user a simple and streamlined way of refining a diagram. For the current version of Foil, I have implemented what is called a "self-organizing map" or Kohonen Neural Network, which is defined as "a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples."[1] In other words, a self-organizing map takes a group of entities which each possess the same set of features and analyzes the similarity of each entity to each other in order to automatically generate a representation of their similarities in terms of those common features. In doing so, a self-organizing map also makes it possible to explore the set of entities that exist between those which were explicitly entered into the system. In Foil, a self-organizing map can be generated for any polygon or geometry within a project and offers a spatial representation of how the different variations of a user's geometry relate to one another. This feature allows the user to easily visualize the variations in his or her ideas across multiple iterations of a design and also offers a quick way for the user to introduce new machine-generated variations into the project.

The current implementation of Foil is just beginning to scratch the surface of what I would like to do with machine-intelligence techniques in the context of digital design tools. As an extension of my work and research on the use of self-organizing maps in Foil, I have spent a great deal of time thinking about Photoshop's "Color Variations" tool and its broader applicability to user interfaces that need to distill complex mathematical representations into easily approachable tools and features. The Color Variations tool is a simple one: it shows the user's image at the center of a grid. Each adjoining image is "More Red," "More Blue" or "More ..." than the center one. When the user clicks one of the adjoining images, the center image adopts the color filtering of the clicked image. This tool presents a very simple, intuitive feedback mechanism - the user clicks what looks good and the image is tailored accordingly. I believe that a hybrid between this sort of tool and a self-organizing map offers an extremely powerful way of thinking about the structuring of a digital design environment and its interface. If we could take all those obtuse menu items from Maya

[1] http://en.wikipedia.org/wiki/Self-organizing_map

and distill them down to a series of choices presented to the user by an interface somewhat like Photoshop's Color Variations tool, then the user would not have to know what an "N-Constraint Membership Tool" does. Instead, the designer would only have to know whether he or she liked the visual effect that tool has upon the project. I believe this idea will be one of the central forces driving Foil's future development. Genetic algorithms also offer many interesting possibilities and I plan to incorporate them into the next version of Foil. In general, my goal for the machine-intelligence aspects of Foil is to develop an artificial intelligence engine that feels less like Big Brother than like a helpful teacher or peer who thinks deeply about your work and then offers some suggestions that may improve your ideas without their becoming someone else's. This idea is of central importance to Foil and is referred to by the project's name. 'Foil' is taken from the literary meaning of that word - a character whose traits contrast with those of the protagonist in such a way that helps to bring the protagonist's characteristics into sharper focus. Foil's name is also in homage to my favorite childhood design medium, aluminum foil.

Personal Statement / Prior Work

I came to ITP with a lifelong interest in virtual reality and three-dimensional graphics. At the age of eight, I began producing a series of aluminum foil mock-ups for virtual reality headsets. This project continued for several years and across numerous design iterations, each one pointed at overcoming a design limitation of the previous. The first was constructed out of aluminum foil, masking tape and a pair of Sony earphones. When this design proved too delicate for the wear and tear of daily usage, a second model was built from a Kleenex box and an aluminum foil frame. This version also added a small cardboard light box and slide carrier to the front of the headset, the first genuine attempt at adding image reproduction capability to my system. Later revisions moved to sturdier, custom-built enclosures as well as more complex optical designs and eventually led to simple electronic circuits such as J-K flip flops, connected in series, to create the simple logic of a sprite advancing from one LED to the next in accordance with a button press. My longterm childhood study of virtual reality hardware has proven to be not only a direct foreshadowing of my later professional interests, but also a key moment in my understanding of the iterative design process. In looking at this work in relation to my high school, undergraduate and graduate work, I am struck by the extent to which the threads of my interest have remained continuous, no matter how disconnected they might have felt to me at various points along the way.

In high school, I moved away from the physical sciences towards Visual Art and Film, and in college, my study of Film and Aesthetic Philosophy led me back to computer graphics. For my undergraduate thesis in Philosophy, I wrote a paper entitled, *"The Photo-Icon: Artistic Representation in the Cinematic Medium of Digital Effects."* One of the central premises of this paper was the notion that the digital special effects industry has focused on what sorts of objects can be placed in front of the camera rather than on the more interesting pursuit of envisioning how the augmentation of the camera (real or virtual) and its way of seeing can contribute to the aesthetic meaning of those objects. After college, my undergraduate background in formal logic afforded me the ability to transition into the study of computer science and I began developing three-dimensional software to apply some of the concepts I had laid out in my theoretical writing. This work has continued at ITP. In my pre-thesis work at ITP, I developed several purpose-specific three-dimensional environments. The most significant of these was DEcomp, a "decompositional environment composer," that was designed to aid in the production of *Digitalis,* a narrative film I created, which used the optical premise of forced perspective in telling the story of a faux deity. At the end of my first year at ITP, I wrote the following about this project:

> While simple forced-perspective transformations could potentially be done manually in a program like Maya, all pre-existing 3D environments take what I call an "object-centric" approach to modeling and as such are not capable of performing complex operations

> within the logic of forced perspective. Early in the process of creating the *Digitalis* narrative, I realized that to make the film, I would need to write my own software that was specifically set up for "perspective-centric" 3D modeling. I began to imagine the set of tools that would become DEcomp – tools that allow a user to stretch and distort the geometry of objects without changing their perspectival appearance for a certain camera position. There are infinite ways that any particular geometry can be manipulated in forced perspective and so I set out to build tools that would not limit the scope of geometric possibilities, but also not overwhelm the user with options. Some of the tools allow the user to graphically manipulate objects while others use a series of drop down menus and nodes to select the desired properties of a transformation.

My thesis work on Foil has been a direct extension of my thinking about the DEcomp and *Digitalis* projects. I find forced perspective to be a fascinating concept and the relationship between DEcomp and Digitalis afforded me an important opportunity in the development of my thinking. That is, by simultaneously developing the idea for a special effects film and the software that would be used to create it, I found that each side of this creative process was greatly enhanced by my work on the other. The film's production guided my decisions about what features should be implemented in the software and my work on and experimentation with those features fed new technical and visual ideas back into my thinking about the film's narrative. Ultimately though, I came to believe that forced perspective, while quite interesting to me, was too narrow an application to warrant a full-featured three-dimensional environment. With Foil, I wanted to apply some of DEcomp's core concepts to more general problems in design.

Conclusion

Three-dimensional design is, in many ways, a complicated matter. To some extent, no software platform can truly overcome this fact. Yet, three-dimensional space is something about which every person has a great deal of experiential knowledge. Through a thoughtfully designed user-interface and with the aid of machine-intelligence techniques, I believe it is possible to distill three-dimensional digital design into a creative and enjoyable process that is accessible to everyone. My process in developing Foil has been much like the creative process that Foil advocates and tries to support. This project represents the latest iteration of my lifelong interest in three-dimensional representation and in systems that aid people in understanding the world around them and the possible worlds they might imagine. Some of the elements that once seemed central now appear tangential, while others that first arose as tangents have since become central elements. I keep coming back to this ongoing project, and each time I reflect upon the previous iterations recorded in my sketchbooks and computers. In Foil, more than in DEcomp or any of the other tools I have developed, I believe I have hit upon a platform that would be of great use to people and I look forward to continuing work on it for years to come.

Research Bibliography

Adobe Systems Incorporated. *Adobe Photoshop Hits Twenty*. Web. 1 Feb. 2011. <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201002/021810PS20Anniversary.html>.

Alpaydin, Ethem. *Introduction to Machine Learning*. MIT, 2010. Print.

AutoCAD WS. Computer software. *Autodesk - AutoCAD WS*. Web. 29 Jan. 2011. <http://autocadws.com>.

Bush, Vannevar. "As We May Think." *The Atlantic Monthly* July 1945: 101-08. Print.

Google Maps. Web. 21 Feb. 2011. <http://maps.google.com/>.

Google SketchUp. Google.

Hiltzik, Michael A. *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*. New York: HarperBusiness, 2007. Print.

"How Many CAD Users Are There? I Mean, Active CAD-using Users (Updated)." Web log post. *WorldCAD Access*. Web. 29 Jan. 2011. <http://worldcadaccess.typepad.com/blog/2010/05/how-many-cad-users-are-there-i-mean-active-cadusing-users.html>.

ILoveSketch. Web. 21 Feb. 2011. <http://www.dgp.toronto.edu/~shbae/ilovesketch.htm>.

Jon Peddie Research. Rep. Web. 29 Jan. 2011. <http://www.jonpeddie.com/publications/cad_report/>.

Kay, Alan. Interactive Telecommunications Program, New York University, New York. 17 Nov. 2010. Lecture.

Kin, Kenrick, Tom Miller, Bjoern Bollensdorff, Tony DeRose, Bjoern Hartmann, and Maneesh Agrawala. *Eden: A Professional Multitouch Tool for Constructing Virtual Organic Environments*. *Pixar Online Library*. Pixar. Web. 21 Feb. 2011. <http://graphics.pixar.com/library/Eden/index.html>.

Autodesk Maya. Computer software. Vers. 2011. Autodesk.

Mitchell, Melanie. *An Introduction to Genetic Algorithms*. Cambridge, MA: Mit, 1999. Print.

NASA Evolvable Systems Group. Web. 1 May 2011. <http://ti.arc.nasa.gov/projects/esg>

Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic, 1980. Print.

SecondLife. Linden Research.

Shapeways. *Shapeways | Passionate about Creating*. Web. 29 Jan. 2011. <http://shapeways.com>.

Vance, Ashlee. "The Wow Factor of 3-D Printing." *New York Time*. 12 Jan. 2011. <http://www.nytimes.com/2011/01/13/technol13ogy/personaltech/13basics.html>.

Zazzle. *Zazzle | Custom T-Shirts, Personalized Gifts, Posters, Art, and More*. Web. 29 Jan. 2011. <http://www.zazzle.com>.

Acknowledgements

This project would not have been possible without my thesis advisors Nancy Hechinger and Marina Zurkow, my ITP advisor Daniel Shiffman, Danny Rozin, Dan O'Sullivan, Eric Rosenthal, Jared Schiffman, David Nolen, Douglas Rushkoff, Heather Dewey-Hagborg, my fellow students at ITP and my family.