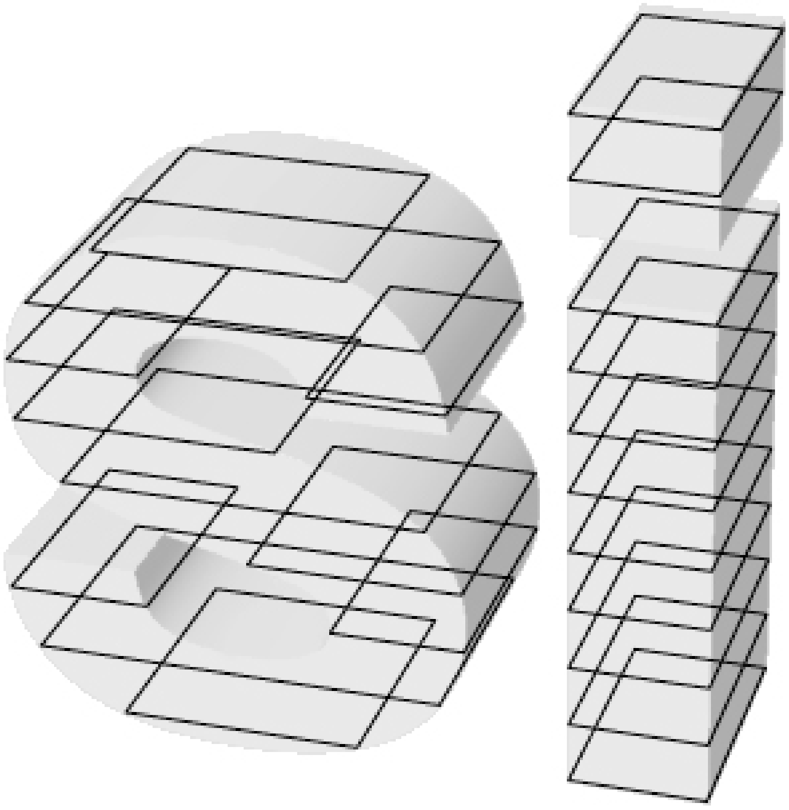
SI 688
Final Term Paper
Analysis of Rhinocerous 3D Modeling System



Michael Senkow

I chose to draw on my previous engineering and architectural experience for this paper. I've used a number of visual programs over the years, both 2D and 3D, both technical and creative. They each use different interfaces, primarily through icons, menus, and keyboard inputs. Rhinocerous 5.0 for Mac stands out from the pack and from watching students in a number of realms I've seen them pick it up faster than other programs. This is due to its use of playing on the potentials of memory, primarily through its visual strengths. I primarily look into how it intelligently used cues to aid in learning and memorizing the system, the positive aesthetic design choices it has made, and how it integrates into a distributed cognitive work frame.

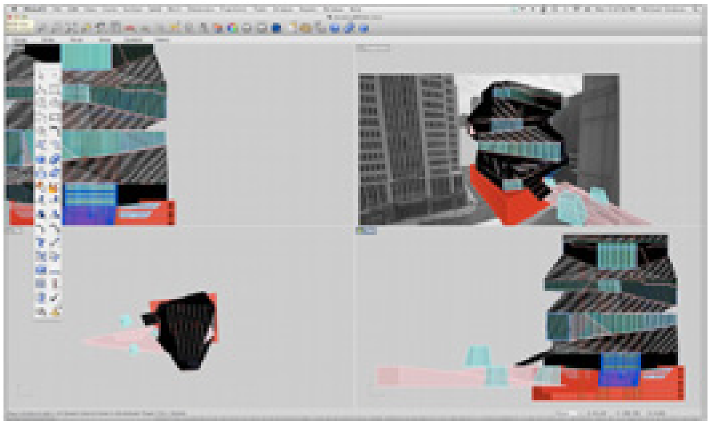


Fig 1. General Example of Rhino Capabilities

SYSTEM

Rhinocerous, generally referred to as Rhino, is a 3D modeling program used in both the design and technical realms. Professions such as industrial design, marine design, graphic design, and digital fabrication professionals all use it in slightly different ways for a wide variety of tasks. On the spectrum of 3D modeling programs, it is both cheaper and quicker to understand than other options (Unigraphics and Solidworks being prohibitively expensive even at student pricing, 3D Studio Max having a collection of inputs based on keyboard entries which requires memorization rather than recognition) but more powerful than many of the free options (Google Sketchup is not as precise and does not have the additional add-ons that allow for parametric modeling.)

At the University of Michigan all the students within the Masters of Architecture program used the program for the creation of 3D models of work, along with rendering these 3D models to resemble real-world objects. Additionally, the add-ons of Monkey and Grasshopper are used to enable this NURBS 3D modeling system to create parametric modeling capabilities. Additionally the program can be useful in the creation of Graphic Design elements. Rhino has the capability of outputting vector files from flattened perspectives of a 3D model, which can then be altered in programs such as Illustrator or AutoCAD. This opens up a user's capability to make complex 2D images of 3D concepts if their artistic abilities are not to the level of drawing them on their own. Strength in this program's usability is how icons and graphics use various visual rules to make a program quick to learn and pull on a user's intuitive understanding of the 3D world to understand what buttons create various 3D shapes and alterations. My intent is to discuss specifically what these visual rules are and how they benefit both the experienced 3D modeling user and someone who is new to the realm.

EXPERIENCE

There is possibly a wide audience for this program. I fall into the realm of someone who has had enough technical background that I was already fairly well acquainted with the program before using it. I went through two set stages in my learning of the program though, that of being the initial new user and then realizing its full strength and potential about half a year later.

My Initial Impression

> *"When I used Rhino for the first time it initially reminded me programs I'd used previously that performed similar functions. Being a 3D modeling program I expected it to do certain tasks that I'd experienced in the past, such as providing different views and creating basic 3D shapes. I scanned the screen that was initially provided and quickly found many of the tasks I would need to do, visually laid out in front of me. The screen was divided into four quadrants, each labeled with a title for the view. I was a little confused at first since the only visual cue to what 'Front' means is a little Z/X axis visual in the lower left. Many other programs used more of a 3D model to display this information. I continued to scan for other information I expected and continued to have my expectations met. Visuals within the tool bar aided me in figuring out quickly which buttons created a sphere, a line, grouped objects, etc. Experimentation showed that actions I'd seen in other programs such as clicking and holding to see more options on a button also held true. There were some options that I did not find but I thought that with time they'd become apparent."*

As time progressed I used the program more and started to explore the different plug-ins available. My tasks stopped being simple creations of shapes and forms and started being complicated mixtures of rendering an image, creating outlines of the object, and outputting these files for use in other programs.

Combining Multiple Actions

“Having more experience with the program I decided to make an interesting visual for the School of Information. I wanted to start with turning some text into a 3D object. I looked around and found the icon that looked like it would implement text, an image of a T, but ran into an issue. The Mac Rhino is still in Beta and this button did not work properly. By this time though I’ve had enough experience that I knew you could type in commands too. I started to type in ‘text’ which popped up. Filling in these options I ended up with a nice looking SI on the screen...but was disappointed to find out that this was not an editable object.

I’d have the options of either trying to outline the image by hand, or import an outline of the text from illustrator. I tried same action, to type in ‘import’ ...and ran into the same issue of it being in Beta and having this not work. Clicking on file at the top brought down a menu though, which had the option to import. Once it was in the document I tried typing COMMAND+T to resize thinking it might have similar options as Photoshop, since other input methods were similar. Sadly this did not work, but there was a resize icon that did. From here I typed ‘extrude’ which automatically brought up a number of other similar options.

Then I wanted to try slicing the object, rendering it, and creating a wireframe to put over the rendered image. Rendering worked the same as previous options, with a simple typing of ‘render’ along with other options in the menu. Contour was also similar. I made the mistake though clicking off the contours I created and being face with the option of selecting each one again (which is tedious) or undoing and redoing my actions. I wanted to try select last but I’d clicked elsewhere already and this option no longer worked. When I typed it in though, one of the options that popped up in the draw-down menu was SelCrv. This implied ‘selection of curves’ and happily it was what I wanted. From here it was easy to output the curves along with the rendered image to make an interesting and unique drawing that combined rendering and outlines.”

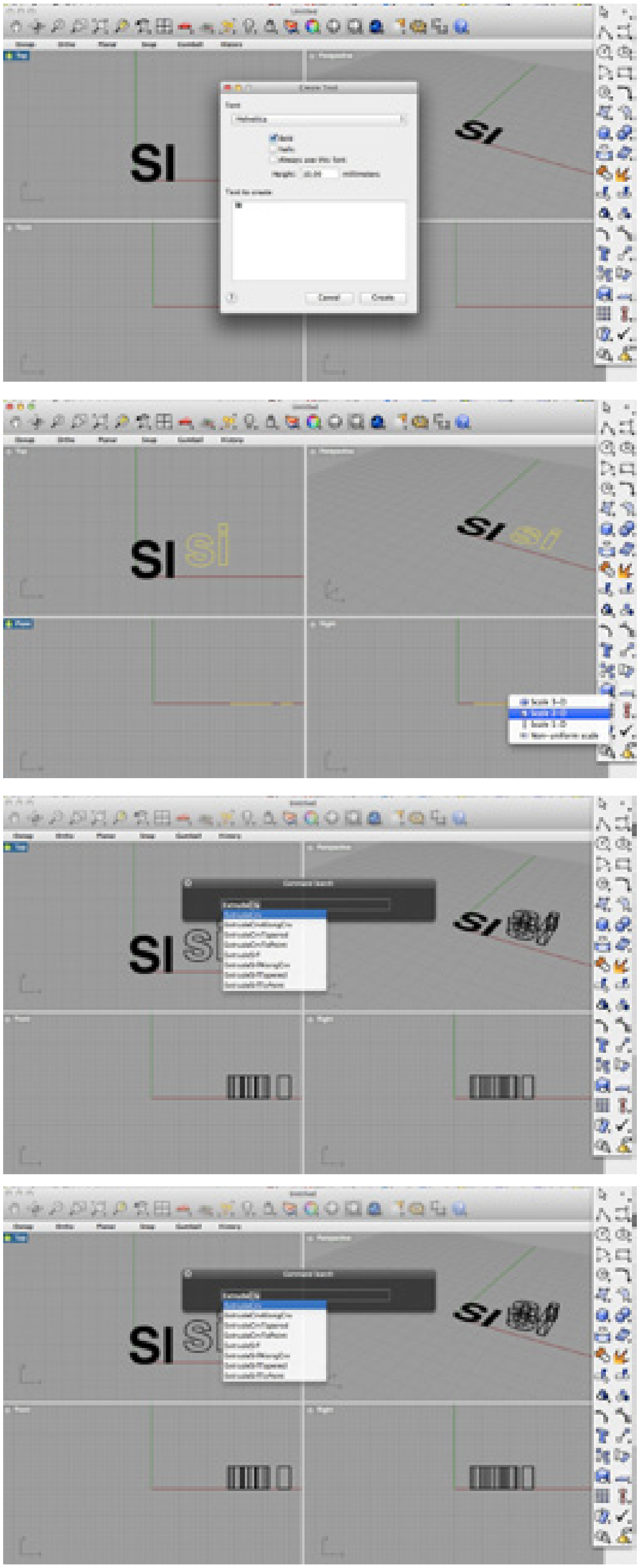


Fig 2. Example 2 displaying the usage of icons, menu choices and keyboard short-cuts, used to create the title page image.

Background/Analysis

I'll be using three concepts discussed in class to look at Rhino, in general attempting to focus more on its use as a tangible medium creation program and how I feel that influences its usability. I'll focus on how it uses long-term and short-term memory focusing on Norman K's learning and transference, positive aesthetics such as those discussed by Norman D to improve the desire for usability, and how the program implements Distributed Cognition to integrate itself into a design environment.

Learning and Transference

There are many forms of human learning and memory, and as seen in the previous section, the human-computer interface impacts them all. In general, learning is the acquisition of knowledge. Within the steps of learning, Norman K discusses that there are different factors that influence recall and transference of information. Stimulus to the sensory systems, positive transfer of knowledge from similar connecting systems, and interference caused by similar systems that have differing actions all impact the ability of the users to understand the new system.

Most of the icons used within Rhino mimic images portrayed both in the analog realm and in other programs. A big draw I believe is the use of recognition of the various icons. This is beneficial more for the advanced computer user, someone who has used numerous computer programs such as Photoshop, AutoCAD, or other 3D modeling programs. As you can see, many of the icons are repeated. Pointer arrows for selection are similar, drawing icons for the 2D shapes are similar, as are other similar actions between the 3D programs (Chamfer edges or resize object) or between the more artistic aspects (Paint object, or move object). There is still a benefit for those who are not as aware of the digital realm though. 'Exploding' and object into separate parts is represented by a spiky, cartoonish explosion shape, whereas joining them back together is represented by puzzle pieces.

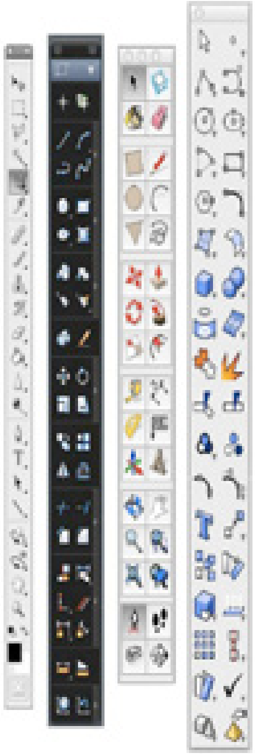


Fig 3.Similarity in icons between Rhino/Adobe/ Autocad

Since recognition is easy and recall is hard it makes sense that the program doesn't try to do any complicated steps in the use of various tools. If anything this directly draws on some of the recommendations made in Designing with the Mind in Mind. Seeing and choosing is easier than remembering and typing. The larger the number of people who use a function, the more visible the function should be so it makes sense that the actions people perform most (pointing, moving, cutting, drawing) are similar across multiple systems. I believe that Rhino actually surpasses other similar modeling programs in its separation of different tasks into different management systems. 3D Studio Max for example is almost entirely keyboard input for all its actions. Solidworks on the other hand has a high amount of point and click menu options, hidden under icons. Rhino plays well at dividing between these systems, increasing learn-ability by putting the common processes in the visual icon realm, but reducing potential error rate by not overcrowding the environment with not commonly used functions.

The manner in which many of these functions are similar to ones on other systems is both negative and positive for the overall learning of the system. Directly within the Rhino program realm, there are very similar properties in both the Mac and Windows version of the program, increasing positive transfer of functionality. There are also many similarities between functions in Rhino and other design programs such as Photoshop or AutoCAD. The problem here is that at times there are false similarities assumed. In my example I mention trying to press COMMAND+T to transform the object, brought on by my experience with Photoshop, but this is a false transference of previous experience.

Aesthetics for Usability

Within The Design of Everyday Things Norman D. discusses various ways in which the aesthetic design of an object influences its usability. Systems and objects should both have good conceptual models, so that the task that is performed makes sense within our expectations. The execution of tasks should also be obvious, and performing a function shouldn't require a myriad of additional hidden tasks for the process to perform properly. Mapping is also key, and goes hand in hand with proper conceptual models and visibility...you don't want to have a conceptual model ruined because the button on the left controls the right hand functions.

Within Rhino the concepts Norman supports are used multiple times. The overall conceptual model used by the program plays into my previous statements on Transference from other programs. The model used for Rhino is very similar to the other design programs

I've mentioned: Menu bars containing the primary new image, open, save, import, etc. are used along with the visual toolbar laid out in a similar function as what is seen in the Adobe Suite. This is a positive aspect of the design... but only if the user has had previous experience with these other programs, and applies their models to Rhino's model. I at first did have a few issues because I acquainted Rhino with the more technical 3D modeling programs, which use interfaces with many more options and far more keyboard shortcuts due to the greater complexity of the programs. It instead shares more with the Design related programs such a the Adobe Suite.

As I've mentioned earlier, visuals are highly important to this program. Along those lines, Rhino has kept them at the front-end of the program, being something you can easily see from the start. Whenever there is a task to be done within the program there are also multiple options for how to actually perform the task. If I want to make a 3D object for example I can either look for the button that resembles the 3D object in question, type in a word that comes close to the shape I want to make, or go through the menu options at the top of the screen looking for the name of the shape I want to create. By having these multiple options at hand the program opens up the potentials for the user to use multiple schema when using the program. I consistently mention my visual connections, but I can easily someone who is more verbal developing just as many positive connections through the typing method.

Mapping is also used well in this program. The initial viewing screen lays out the 3D object in four views. The user has the option of placing whichever views they wish to into these, but the program starts off placing the right view on the lower right and the top view on the top left. Its choice of putting the front view on the lower left could be questioned, but with so many viewing options I have a hard time seeing users being confused by this one choice. Many of the other uses of Mapping are akin to what you'd see in other programs: arrows being placed on the button options pointing downwards, to key in the users that there are more options or a box appearing showing the selection area when the user is clicking the mouse button to select multiple objects.

Distributed Cognition

Very few computer systems now-a-days are stand-alone systems. Most are computer supported cooperative work (CSCW) systems, in which the computer itself is only a tool that is part of the overall process. Each of the workers in the environment must be considered, along with the other programs that work with this system. In this conceptual framework, Rhino can be considered just another processing unit as part of the overall system.

Rhino uses this concept both in the program itself and in its overall connection to the design community. It has not quite broached the subject as well as I've seen other 3D modeling programs commit to distributed work though. Unigraphics, a more technical program, has methods in which multiple users can check in and out their 3D models, allowing many people to work on the final 3D object by dividing up the work among many. Rhino does not have this capability, but it does integrate with other design programs very well. Files can be exported into different formats, readable from everything from Illustrator to AutoCAD. This opens up the potential of someone working on the Rhino 3D model and finished up the final touches, when it's been exported into an Illustrator file for line-work to be finished up and at the same have the entire final document laid out via a tool such as InDesign.

Beyond this direct connection of the program integrating with other programs, there is also the way in which the company has opened itself up to the wider community of developers. Companies such as Adobe or AutoCAD are very closed to the public in how the programs are being developed. Rhino though, while not a fully open-source piece of software, has multiple ways in which the program opens itself to the public. With the Monkey plug-in, users are able to create new tools for the program that do not currently exist. For example, in my use of the program there is a Pipe command, which lets you run a cylinder along any line that has been drawn. There is no Pipe-All command for multiple lines though. With the use of Monkey though, I was able to write a script that took all the lines and Piped them. With this ability to add features, multiple on-line forums have opened up, in which users share the programs that they have created, so that others can perform similar tasks. This continues the idea that Rhino is only one part of the greater whole, being a singular node in the creative process and integrating with the wider community very well.

Failings in Rhino

There is a section of Rhino that goes against my base statement that Rhino has an easy to use interface due to its visual cues. When the program becomes more complicated, or starts using options that are harder to create a visual reference to a keyboard typed command is required and used. While generally most of what a 3D modeling program does can be represented by visual concepts, how would 'Select all Surfaces' be represented? Or 'Select Last'? Or while the concept of 'Show Edges' may make it seem like a good visual icon, what it actually is used for is to show Hidden Edges. These more complicated processes need the more complicated keyboard input.

Initially these were not something that came naturally to me, but I've realized after using the program more that the keyboard interface could be beneficial for those with a programming background. Many of the typed selections are phrased in the same way that I've seen methods and variables in a program phrased.

The program also has few areas in which the aesthetics providing positive design interface could be improved. I was going to discuss icons that could be improved, but looking through them I found very little by way of design that could be improved to make them more understandable...though this may be biased by my having a background that used similar programs and having used the program myself for a while. The main realm I can see it improving aesthetically is by integrating more keyboard shortcuts similar to what the Adobe Suite has. It already has models that appeal to those who use icons, those who use menus, and those who type in commands. Adding in shortcuts would be the final improvement.

Last of all, Rhino should integrate the distributed work frame that I've seen in other 3D modeling programs. If it allowed for multiple users to work on the same 3D model more readily it could easily surpass the strengths of its more-expensive counterparts. It would also continue to build upon the Distributed work frame it has already started to set down.

Conclusion

When I first started using the program I approached it as I had Photoshop or Solidworks. After having used it though I discovered the more complex uses within the program. Over time I even started using the scripting additions that allow for more complex alterations of shapes. Much of the way in which I've approached Rhino would have been altered if I'd started with a different background when using it. I feel though that overall the company producing the software has tried to create a program that is easily used by a wide variety of audiences, appealing to multiple forms of memory and learning enabling, using different conceptual models on how to use the program, and integrating its help systems and files to be easily connect many users.

References

Norman K. L. 2008. Ch. 6 Learning and Memory, Transfer and Interference. In Cyberpsychology: An Introduction to Human-Computer Interaction. Cambridge University Press. New York, New York.

Johnson, J. 2010. Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. Morgan Kaufmann, San Francisco, CA. Ch. 7, Ch. 9

Norman, D. 1988. Ch1+2: The Psychopathology of Everyday Things and The Psychology of Everyday Actions. In The Design of Everyday Things. Doubleday. New York

Perry, M. 2003. Distributed Cognition. In J. M. Carroll (Ed.) HCI models, theories, and frameworks: Toward a multidisciplinary science. Morgan Kaufmann. San Francisco, CA.