Croquet: A Menagerie of New User Interfaces

David A. Smith
104 So. Tamilynn Cr.
Cary NC, 27513
davidasmith@
bellsouth.net

Andreas Raab
University of
Magdeburg, Germany
andreas.raab@
squeakland.org

David P. Reed[1]
HP Fellow
Hewlett Packard
Laboratories
One Cambridge Center,
12th floor
Cambridge, MA 02139
dpreed@reed.com

Alan Kay[2]
Senior Fellow
Hewlett Packard
1209 Grand Central
Ave
Glendale, CA 91201
alan.kay@
viewpointsresearch.org

ABSTRACT
A new architecture like Croquet presents numerous opportunities and challenges to create useful interfaces to enable access to the underlying power of the system. In particular, our focus on an integrated 2D and 3D system ensures that we have a rich intellectual environment within which to explore. This experience is similar to the development of the original modern windowing user interface created by Alan Kay, his team at Xerox Parc, and his Squeak team[3,4]. Just as those teams did, we also have an infrastructure that is rich enough to allow us to perform some deep exploration into the presentation and manipulation of rich media expressed in 3D, and in the negotiations through these rich media worlds.

A number of key technologies will be discussed including dynamic movable portals, floating 3D windows, true 3D creation tools, live teleporting snapshots, and completely new concepts such as 3D portals.

KEYWORDS
Croquet, collaboration, User Interface, 3D graphics, Squeak, Smalltalk, TeaTime, OpenGL, peer-to-peer, 3D portals.

INTRODUCTION

Croquet [11,12] was built from the ground up with a focus on deep collaboration between teams of users. The system is focused on interactions inside of a 3D shared space that is used for context-based collaboration, where each user can see all of the others and their current focus as well as interact with them. Of course 2D is considered to be a true subset of the 3D environment, and 2D interaction is utilized throughout the system.

This richly collaborative environment presents both an opportunity and a challenge to the user interface designer. By default, all of the interesting objects inside of Croquet are immediately collaborative – that is, they are easily shared, allowing multiple users to interact with them simultaneously.

Further, the fact that all of the objects exist in a shared 3D environment forces the designer to consider issues relating to 3D orientation, as the user can approach an object from virtually any direction; and scale, because the user can be any distance away from an object.

This is by no means a complete description of either the current state of the Croquet UI, nor the potential of interactive 3D interfaces. It is primarily a list of useful objects and techniques that we have developed over the short history of the project. A key part of the Croquet architecture is that the user interface is just another collection of objects that can easily be replaced or enhanced. A fundamental idea behind Croquet was to develop a flexible enough framework that virtually any UI concept could quickly and easily be prototyped and deployed. Our attempts here can simply be thought of as examples of the potential of the system.

RELATED WORK

Early Work

Sutherland's work [15,16] on direct manipulation and modeling of graphical object based entities established the first true steps toward an interactive human computer user experience. Not only did he establish a great deal of the fundamental methods for how to create and manipulate interactive environments that are still quite relevant, but his focus was on creating a tool that would fundamentally amplify human capabilities. He further extended the potential of human interface with his development of the first tools for the exploration of virtual environments.

Krueger [5] created a number of impressive collaborative demonstrations in his Videoplace. The level of interactivity and intimacy was so strong that when one user would reach the image of his hand out to touch another, the other user would pull away. This was the first demonstration of shared virtual objects where multiple users could simultaneously interact with their environment and with each other.

---

1 Dr. Reed is also an Adjunct Professor at the MIT Media Laboratory. His ongoing work on Croquet is supported by Hewlett Packard, and correspondence regarding Croquet should be directed there.

2 Dr. Kay is also President of Viewpoints Research Institute, Inc.

Movement in 3D Spaces

The Colony [9] was one of the first immersive 3D adventure games on a PC. Later, Virtus Walkthrough [10] also utilized this interface. Croquet continues to utilize the method of moving through a space using the mouse pointer that was invented for the game.

3D Simple Manipulation

The Virtual Trackball demonstrated a simple method to manipulate 3D objects for viewing. It was defined by a fixed invisible sphere around a 3D object. When selected, the sphere and its contents tracked the mouse pointer position such that the original point followed the mouse exactly. This allowed the user to quickly and easily access any part of the object. The problem with this approach was that the user had no control over the orientation around the axis defined by the mouse position. This was somewhat resolved by adding a circular area around the track sphere that allowed rotations around the axis perpendicular to the screen.

Seidl [8] developed a 3D manipulation cube which greatly enhanced and simplified the manipulation of 3D objects in a space. When an object is selected, a pointer sensitive cube is generated which completely contains the object. When the user clicks and drags a face of the cube, the cube and the enclosed object translate parallel to the selected face. When the user selects an edge of the cube, the object rotates around the axis parallel to the selected edge, and when the user selects a corner of the cube, the object is scaled in or out depending upon the pointer's distance from the center of the cube.

3D Systems

Smith's work on ICE – the Interactive Collaboration Environment, a multi-user shared component environment and later the Virtus OpenSpace architecture [11] acted as an important guide to the resulting Croquet system and in a sense Croquet is a far more complete result of this work.

Fisher et al [2] developed a powerful, totally immersive 3D working environment. This system included the ability to dynamically interact with the system via 3D menus and window documents, and the ability of the user to directly manipulate his position and orientation inside the world and interact with the objects that inhabited it. Further, the system could interact with the user as if he were just another object inhabiting the space. The best example of this was the virtual escalator that the user could step on that would then carry him up to another floor.

3D Design

Virtus Walkthrough [10] continued the interface work done for the Colony by allowing users to design a virtual environment and then immediately experience it as it was being built. Walkthrough used a generalized extrusion model for creating geometric objects that was extremely simple to use, but allowed the user an enormous degree of flexibility. Virtus Walkthrough was also the first product to incorporate a virtual track ball to manipulate 3D objects.

OVERVIEW

We have a number of goals in developing interactive 3D user interfaces in Croquet. First is to achieve the ideal of the invisible interface. This does not actually mean that the elements of the interface are not visible, but that the user will quickly learn how to achieve his goals with minimal effort. That is, the user ceases to see the interface itself, but is able to concentrate on achieving the desired results.

A second goal is to build interfaces that scale well with multiple users. This means ensuring that every user action (within reason) is visible to all of the other users. A user's focus of interest is communicated with his position and orientation in the 3D world, and the position of his pointer in the world. Further, the changes to the object that the user is working with are clearly visible to everyone else as the user manipulates it.

A third goal is to take advantage of the existing infrastructure of hardware capabilities. Hence, we have focused on standard PCs with display devices and keyboard and mouse input devices. In particular, this document will focus on mouse driven input and control. This could be considered to be more of a restriction than a goal, but the mouse has proven itself to be an extraordinarily useful device over its long lifetime. It has a high degree of control with minimal effort that is ideal for interacting with 3D environments for long periods of time. Its usefulness in this context is still being explored. In any case, the actual object that represents the user's input in our 3D space is an actual 3D pointer which could just as easily be controlled by a 3D tracking device. Our hope is that Croquet will be useful to a large population, which precludes any kind of serious modification or enhancement to the current systems, at least for the foreseeable future. Thus, the primary focus of this paper is on mouse/pointer driven 3D UI.

A fourth goal is to move as much of the decision making of how an object responds to the user into the object itself. Each object defines its own interface with the user, so in a real sense, this discussion is focused more on the object interface to the user.

Much of the current interface to Croquet is discussed in [12] and in [13]. This document attempts to clarify and formalize some of the decisions that were made in developing the current interface, as well as describing some of the important extensions we have developed.

PICKING AND MANIPULATING

When we consider the user interface to the Croquet virtual environment, we need to look at two orthogonal elements – the movement of the user through the space, and the user directly interacting with the objects that inhabit it. Given the limitations we have imposed on extending the basic PC, we have resorted to a strategy that allows us to change the modality of a standard three button mouse depending upon which button the user selects. Given that we expect the keyboard to be used primarily as a text input and manipulation tool, we have also attempted to limit its use in

traversing a space. In particular, we have avoided the use of the "mouse look" method of traversing a space precisely because of this reliance on dedicating the keyboard to motion and simple interaction. Of course, that does not mean that it cannot be used in this way.

Currently, Croquet uses the left mouse button to select and manipulate, and the right mouse button for movement through a virtual world.

MOVEMENT

The main method of moving through a Croquet space is well described in the current draft manual[12] and in Smith[13]. Two other UI methods useful for moving through a space have been developed.

Mouse Wheel up/down

Sometimes it is necessary for the user to move up and down in a space. Most spaces have a solid "floor", which the user camera is constantly testing against. In a sense, the user is constantly falling to the floor. However, when there is no floor under the user, or he needs to jump up or down, the mouse wheel can be used. Simply rolling it forward moves the user down, and rolling it back moves the user up. If there is a floor under the user, the wheel can be used to jump up onto objects that are two high for him to get to otherwise.

SnapTo/SnapBack

Distance is a very useful concept for managing large amounts of information. Objects are iconized by distance. This means that the icon and the object are the same thing. The cognitive distance between the two is effectively zero. A house seen far away is obviously still a house.

Although distance doesn't really mean much in a virtual environment, it can be inconvenient to have to virtually "walk" between two objects that are far away. What we really want to be able to do is instantly jump to an object that is far away, and if we find it of no interest, immediately jump back. Currently, the 3D windows in Croquet support this capability. If a user is far away, he can click on the arrow floating over the top of a window or right mouse quick-click anywhere on the window itself. He is then quickly snapped to an appropriate position in front of the window. If he finds that this window is of no interest, he can quick-click again and he will be snapped back to his original position. To ensure that the user doesn't get lost in the transition, an interpolated path (position and orientation) is computed between the start and end poses. The effect is a zoom to (and away) from the object ensuring that the user understands what is happening and where he is going.

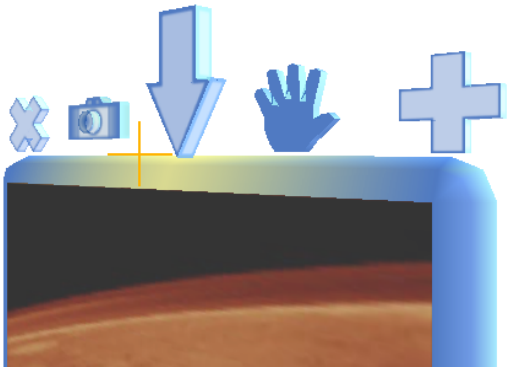
This same model can be utilized to bring objects to the user and return them to their original position.

PRIMITIVE INTERFACE CONTROLS

The rest of the user interfaces that will be described here are focused upon manipulation and creation. All of these are built out of a few simple primitive interfaces that are described here.

Buttons

Buttons are the simplest interface from the user's point of view. In the case of Croquet, buttons are simply objects that when clicked send a message to a particular target. Buttons can have a single state, where it sends only one kind of message, or can act as a switch, for example to open and close a window portal.



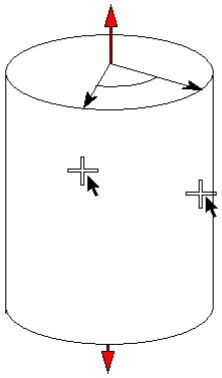
Here are some examples of 3D buttons found over the windows of Croquet.

Spherical Spinners

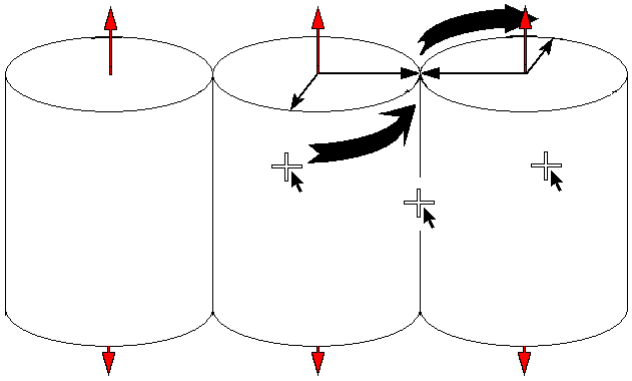
Spherical spinners are also referred to as virtual track balls. This interface was originally invented by Chen[1]. The user selects a point on an object he wishes to spin. This point is used to define a virtual sphere that has a radius defined from this selected point to the center of the object. As the user drags the mouse across the surface of the virtual sphere the object is reoriented such that the original point is always directly under the pointer. This interface is extremely easy to use, and though the user has complete control of the final orientation of the original selected point, it has the disadvantage that it is difficult to control the rotation of the object around the line defined by the selected point and the center. Chen added an additional degree of control with a ring around the virtual track ball, but this is impractical in a 3D environment. Instead we have focused on a more constrained interface.

Cylindrical Spinners

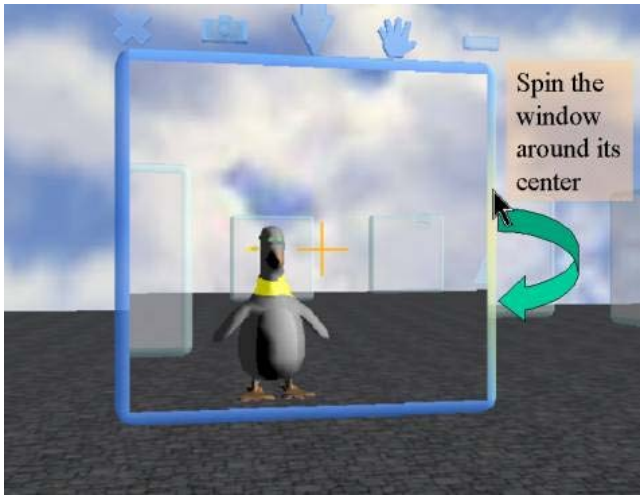
Cylindrical spinners are similar to spherical spinners in how they are used, except that rotation is constrained around a predefined axis. When the user selects the object, a radius is computed between the selected point and this axis. A virtual right cylinder is computed and the position of the pointer is tracked on this cylinder. As the user drags along the cylinder, the angle between his original point and the new point is computed, and the object is rotated around the axis. Translations along the axis are ignored and the cylinder is considered to be infinitely long.



This approach only allows for a maximum of 180 degrees of rotation. Further, if the user were to select a point that was near the visible edge of the cylinder, he could only rotate the object in the other direction. To allow the user to rotate the object as much as a full 360 degrees in any direction, we add additional virtual cylinders of the same size and parallel to the original, but with a reverse of the rotation. As the user drags across one cylinder to the next, the angle of rotation continues to be added to the object, but it is the negative of the value computed from the rotation of the second cylinder.



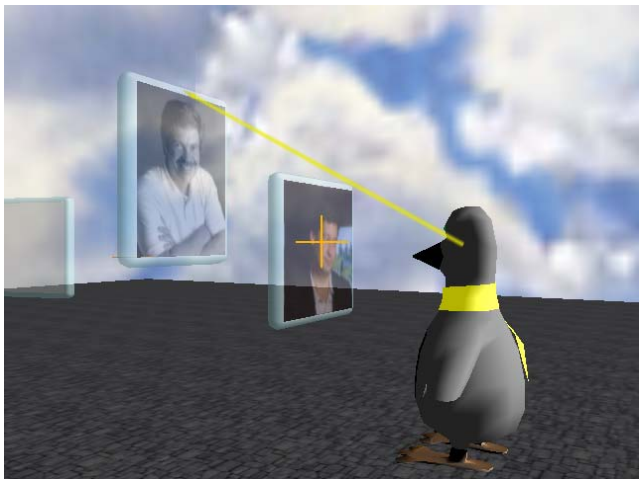
The 3D windows in Croquet utilize this rotation mechanism. The user can select the left or right sides of the window frame and rotate it around its central axis.



This image displays an example of the cylindrical cylinder in action. When the user selects the edge of the 3D Croquet window, a virtual cylinder is computed and the object can be rotated around its up and down axis.

Planar Translations

Planar translations allow a user to move an object constrained to a specified plane. This plane is defined based upon the point of the user's selection and a normal vector defining the orientation of the plane. As the user drags the pointer across the plane, the object is translated to the same point. This method is used to drag the Croquet 3D windows in a plane that is defined to be perpendicular to the orientation of the user. The manipulated object can also define this plane.



The user is dragging the window in a plane that is perpendicular to the users orientation.

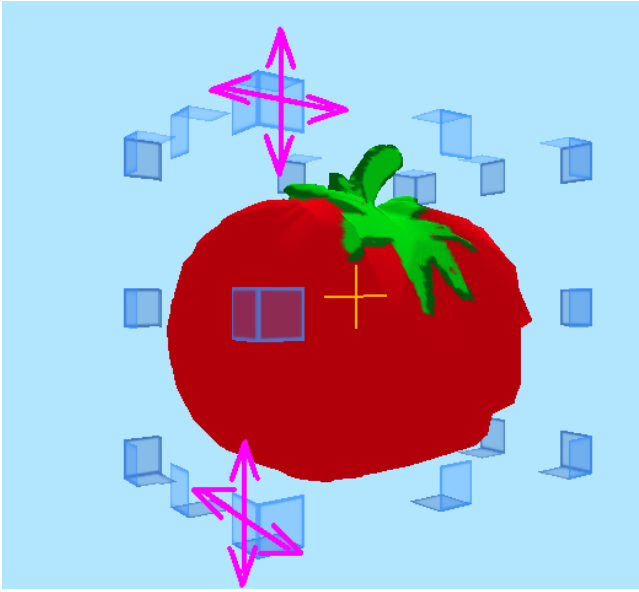
COMPLEX INTERFACE CONTROLS

This next group of controls is basically built upon the primitives discussed above. They are really compound interfaces where each of the primitive control tasks has a well defined and specific function inside the larger scope of the interface. The first control is a 3D edit box that allows

extremely simple positioning and orientation of complex objects in a 3D space. The next is a fundamentally new concept called a 3D portal. This is a miniature live version of a space, which can even be embedded in the original space. The final discussion of this section is a new approach to 3D computer aided design that allows the user to quickly create and edit even complex organic shapes.

3D Edit Box

The 3D edit box is used to manipulate and position objects in the 3D environment. It is used to translate the object parallel to any of the faces of the box, rotate around any of the three axes, and scale it in and out from the center of the box. The side of the box is made up of eight squares, four in the corners, and four on the edges. Three sides meet at each corner of a box, so in each corner we see three rectangles. Two sides meet at each edge of a box, so we see two rectangles meeting at each side of the box.

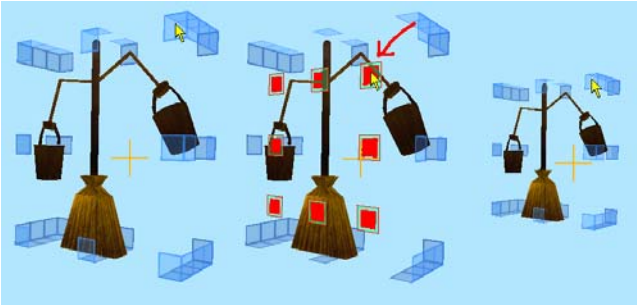


In this image, we see the edit cube. If the user selects one of the corner rectangles in the edit cube, the object will be translated in a direction parallel to the surface of that corner rectangle. The arrows above indicate the directions that the object can be dragged when the underlying rectangle is selected. This allows the object to easily be positioned virtually anywhere in the 3D space. This utilizes the planar translation primitive interface control.

When the user selects one of the edge rectangles, as shown, the object is rotated around the axis parallel to the selected edge.

This rotation is performed in the frame of reference of the object itself. This makes it trivial to use. This rotation uses the cylindrical spinner primitive interface control.

Complex, hierarchical objects can be easily edited as well. Once the object is selected, the children of the object can be selected by pressing a down arrow. Sibling objects can be selected by pressing the left or right arrows, and the parent can be selected by pressing an up arrow. Once selected, each of the individual children can be edited in exactly the same way as the above picture illustrates.



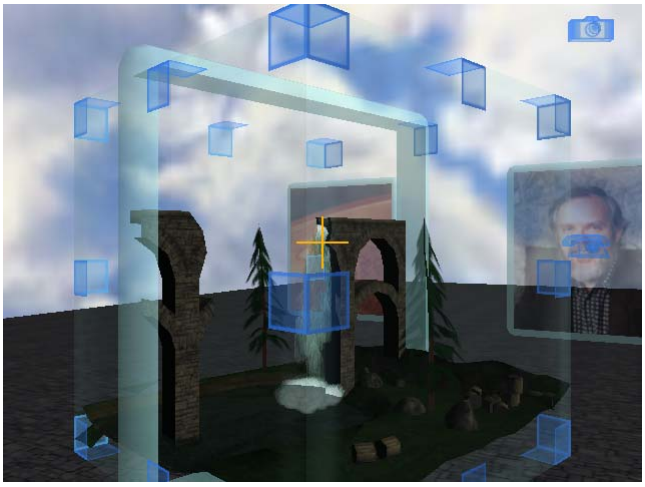
The corner rectangles can also be used to scale the object. If the corners are selected while the shift key is down, then the object is scaled proportionately by the distance of the cursor from the center of the edit cube.

The Croquet edit box is a direct enhancement of Seidl's work[8]. One of the key problems with the earlier effort was that if the enclosed object had any functional elements, they

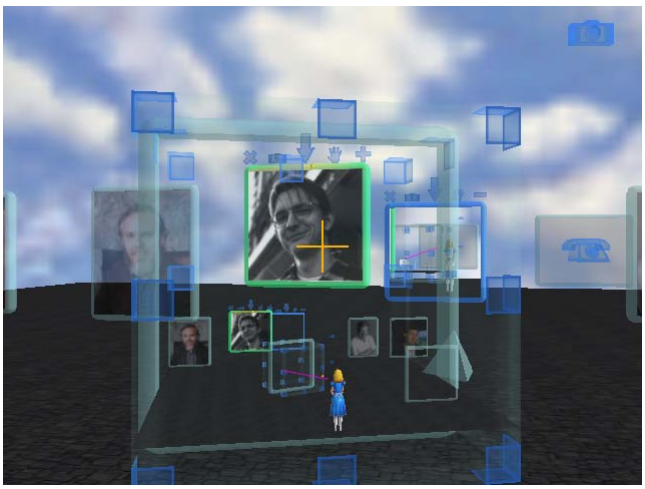
were unavailable while editing. The Croquet edit box has all of the editing capabilities of the earlier effort, but allows complete access to the enclosed object's function.

3D Portals

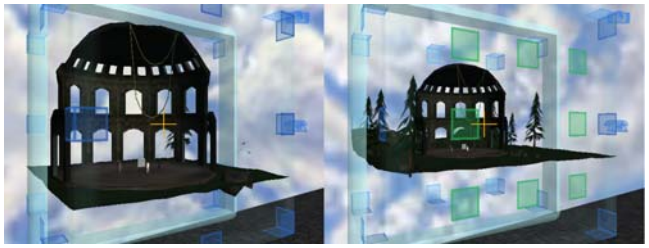
3D portals are a new concept in interactive visualization and design. These are complete miniature copies of existing spaces that are completely accessible and editable by the user. They are inside of a clipping box, which can in turn be inside of another 3D edit box. 3D portals are extremely useful for getting a complete overview of an environment, even when the user is inside that same environment.



This is an example of a 3D portal inside of an edit box. The edit box has the same properties as described above, except that instead of translating the entire contents of the box, the corner rectangles are used to scroll through the 3D space.



Here we see a miniature version of the same space that the user is in. Notice that the windows inside the smaller 3D portal are identical to the outside environment. The girl in the blue dress is actually a miniature version of the user's avatar. The user is moving the window in the center of screen by manipulating it in the 3D portal, but the full-size version is moving in exactly the same way. This is simply because it is really the same object.

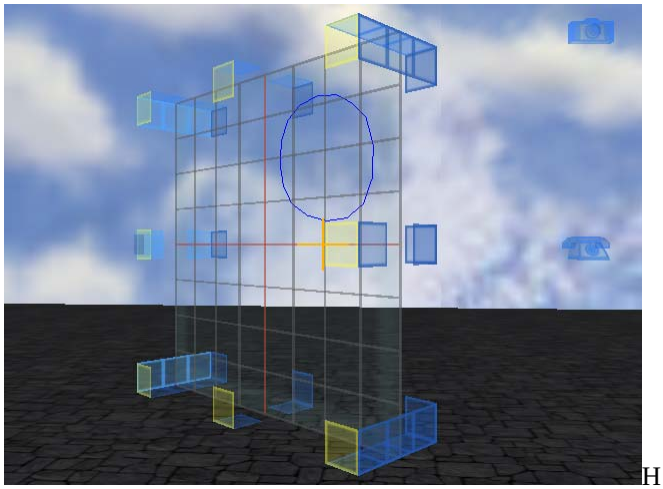


The above image demonstrates how the 3D portal allows the world can be scaled using the 3D edit box scaling rectangles.

A key advantage of 3D portals is that because they are always live working environments, they allow for a god's eye view editing capability that is simply unavailable in traditional approaches to design. The user can edit a miniature version of an object in the space while experiencing the results of the edit in the full-size space.

Wicket – 3D CAD

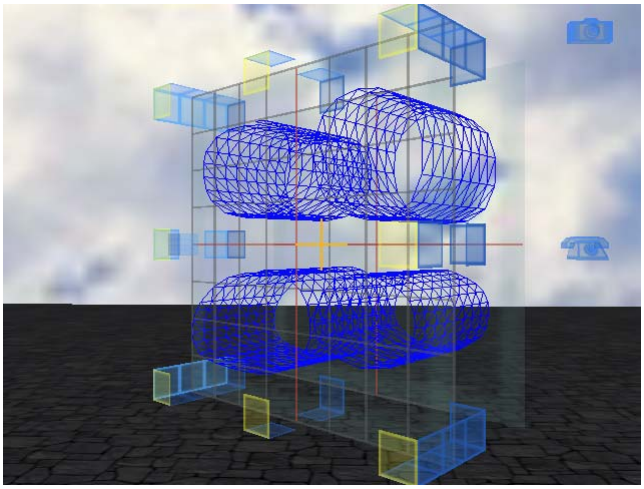
Wicket is a free-form, spline-based, 3D design tool. The goal is to provide the user with a flexible combination of drawing and sculpting tools. The primary interface is a 2D drawing surface, which is contained in a standard 3D edit box. The user can simply draw a 2D shape on the drawing surface, and this shape is immediately converted to a spline path.



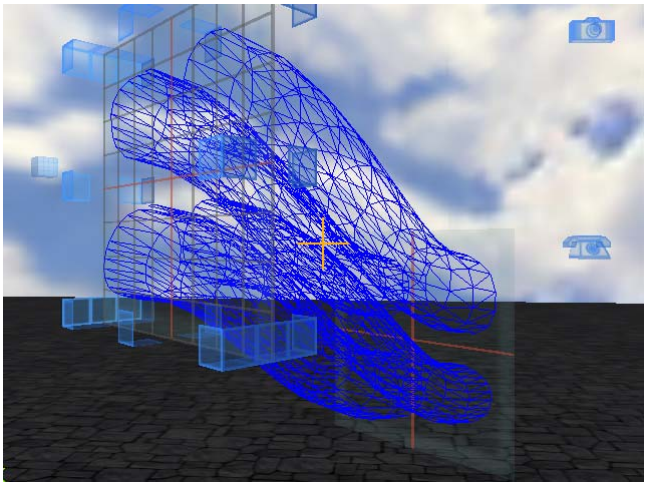
Here the user has drawn a circle. A number of spline paths can be created on any given surface. These new spline paths are still quite editable, and are actually represented by 3D points themselves. This means that they can even be pulled off of the surface if necessary.

Later implementations of Wicket will include a gesture based interface that will allow the user the ability to create perfect circles, rectangles and lines.

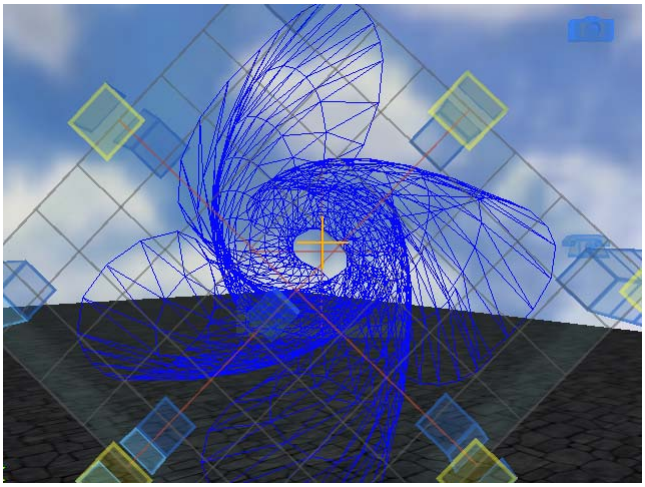
Once the user has specified a base outline path as above, he can click on the small cube button floating over the surface to extrude the path.



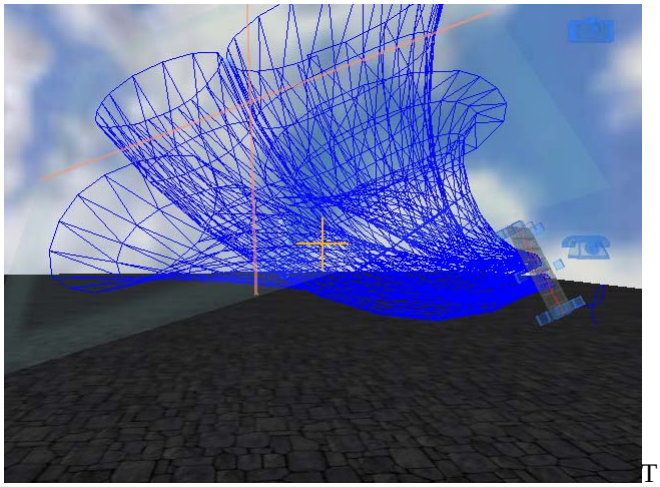
When the spline paths are extruded, a new edit surface is generated, and the spline paths themselves are used to generate a spline surface.



If the user drags the new edit surface, the spline surfaces are dynamically recalculated. Additional control surfaces are generated to ensure a smooth curved surface.



The user can twist the edit surface, and appropriately interpolated spline curves will be generated. The surface edit box also allows scaling of the spline paths it contains.



This image demonstrates the construction of a complex object after scaling, rotation, and translation of the two controlling edit surfaces. Notice the grid and edit box are removed from the inactive edit surface. These edit boxes are exactly the same as that described above.

Wicket represents a good start in a fundamentally new approach to computer-aided design. A number of additional design elements have not yet been implemented. One of the more interesting ones is the use of gestures to specify outlines. The user will be able to quickly sketch a circle, rectangle, or even complex shape, and the system will generate a perfect version. Commands can be specified in the same way.

Currently, the extrusion model is a simple spline, but any continuous curve can be used to generate surfaces. The user should even be able to specify a mathematical function for the path to follow. The edit surface should support editable text to allow the user to enter this information directly.

An editable button set will be added that allows the user to impose constraints and micro-modes, and create more standard kinds of interfaces to the system. One design goal is that these buttons will be created with Wicket, and then the button's behaviors will be created with Croquet's scripting language.

The user will also have the ability to use any surface, even a curved spline surface, as an editable surface. This means that it will be extremely simple to generated extremely complex organic shapes in a matter of minutes.

The key to a powerful CAD system is that it should allow the user to sketch out a rough design extremely quickly, and then allow for constant refinement and tweaking. The user should always feel comfortable about being able to throw ideas away when they don't work, because he has spent a minimal amount of time on it. Wicket needs to support this process.

FURTHER WORK

Many of these interfaces are still preliminary designs. In particular, Wicket has only scratched the surface of the potential of this approach to CAD. The goal is to develop an industrial quality system that will allow teams of users to work together in the design process. In a sense, Wicket is the ultimate example of the power of deep collaboration in Croquet.

The 3D portal is a stunningly simple, yet powerful concept. However, the very fact that it is a fully functioning miniature version of a 3D world means that the interfaces to the objects contained within are themselves miniaturized. A strategy for dealing with this needs to be developed to properly tap this powerful approach to design.

A collaboration environment requires people management tools. An interface needs to be developed that displays areas of interest available on the net when Croquet first starts up. Given that a user can only view a small area at any given time, and that the other users of interest may not be directly visible, visual interfaces need to be developed that quickly show the user where others are and their current context.

Sound is a crucial part of any immersive environment. It has huge value for developing new classes of sound based interfaces, for feedback and information. We are studying the developing industry standards for 3D sound. In particular, the work in OpenAL [6] seems to be promising.

CONCLUSION

Croquet has been designed from the ground up with a focus on enabling large scale peer-to-peer collaboration inside of a compelling shared 3D environment. We have found that the existence of such a powerful development environment allows us to deeply explore new ideas in interface design that were simply impractical to even consider in the past. Much of our efforts are more akin to exploration than invention. New ideas always seem to lead directly to the next set of problems that themselves have natural solutions. On the other hand, this is a new kind of environment that requires a good bit of effort from the user interface designer to really understand how to maximize the users access and power. We are still just learning what is possible here.

ACKNOWLEDGMENTS

We would like to thank Kim Rose, Michael Rueger, and Yoshiki Ohshima for their help and insight. The Croquet project was generously supported by Hewlett Packard Corporation, and Applied Minds, Inc.

REFERENCES

1. Chen, M. *A study in interactive 3-d rotation using 2-d control devices*. Computer Graphics (SIGGRAPH '88 Proceedings), 22(4):121--129, 1988.
2. Fisher, Scott S., James Humphries, Michael McGreevy, Warren Robinett, “The Virtual Environment Display System,” *1986 ACM Workshop on Interactive 3D Graphics.*
3. Ingalls, Dan, Ted Kaehler, John Maloney, Scott Wallace, Alan Kay. Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself, in *Squeak: OpenPersonal Computing and Multimedia* ed. Guzdial, Mark and Kim Rose. Prentice Hall, 2002
4. Kay, Alan. "The Early History of SmallTalk", in Bergin, Jr., T.J., and R.G. Gibson. *History of Programming Languages - II*, ACM Press, New York NY, and Addison-Wesley Publ. Co., Reading MA 1996, pp. 511-578.
5. Krueger, Myron W. *Artificial Reality II*. Addison-Wesley. 1991.
6. OpenAL Web Site: http://www.openal.org/home/.
7. OpenGL Architecture Review Board, Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, *OpenGL Programming Guide*, Third Edition. Addison-Wesley. 1999.
8. Seidl, Robert *Object-oriented curve manipulation system. US Patent* 5,583,977. Dec 10, 1996.
9. Smith, David A. The Colony. Mindscape 1987.
10. Smith, David A. Virtus WalkThrough. Virtus Corporation. 1990.
11. Smith, David A. *OpenSpace: A Small System Architecture*. Virtus Corporation. Unpublished Document. 1994.
12. Smith, David A., Andreas Raab, David P. Reed, Alan Kay. *Croquet User Manual* v.0.01 Web site: http://www.opencroquet.org..
13. Smith, David A., Alan Kay, Andreas Raab, David P. Reed. *Croquet – A Collaboration System Architecture*. C5: Conference on Creating, Connecting and Collaborating through Computing. 2003. p.2-.
14. Squeak Web Site: http://www.squeak.org/
15. Sutherland, Ivan. *Sketchpad: A Man-Machine Graphical Communications System*, MIT PhD Thesis, 1963.
16. Sutherland, Ivan. “The Ultimate Display”., *Proceedings of IFIPS Congress 1965*, New York, New York, May 1965, Vol. 2, pp. 506-508.