THREE-DIMENSIONAL VIRTUAL ENVIRONMENT

FOR SPATIAL DEVELOPMENT

A Thesis

by

KATHLEEN SUZANNE BATEMAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Visualization Sciences

THREE-DIMENSIONAL VIRTUAL ENVIRONMENT

FOR SPATIAL DEVELOPMENT

A Thesis

by

KATHLEEN SUZANNE BATEMAN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Frederic I. Parke (Chair of Committee)	Louis G. Tassinary (Member)
Lauren D. Cifuentes (Member)	Phillip J. Tabb (Head of Department)

December 2004

Major Subject: Visualization Sciences

ABSTRACT

Three-Dimensional Virtual Environment for Spatial Development. (December 2004)

Kathleen Suzanne Bateman, B.A., Grinnell College

Chair of Advisory Committee: Dr. Frederic I. Parke

The purpose of this project is to design a computer program to help children understand a strategy for changing vantage points within an imagined space and gain knowledge of how spatial transformations work. The developed software, called Viewpoints, presents a virtual three-dimensional environment to be explored and modified by the user. Object and camera manipulations are illustrated through animation. Furthermore, the program was designed to have an intuitive interface and be easy to access. This should allow the software's target audience of children to focus on the spatial orientation and spatial visualization aspects. A small study evaluated the software in terms of content, instructional design, technical quality, student use, and instructor use. The study provided valuable feedback on how to improve Viewpoints in the future. Information gathered suggests the issue of speed should be addressed and supplemental materials should be added.

To Leif, for everything.

ACKNOWLEDGMENTS

Many thanks to my committee for their guidance and support:

- Dr. Frederic Parke, Dr. Louis Tassinary, and Dr. Lauren Cifuentes.

Much appreciation also goes to:

- Leif Brown for many hours of encouragement, help with design and organization, answering questions big and small, financial support, and most especially love.
- Robert and Leslie Bateman for help with the activity booklet, program testing, editorial review, moral and financial support.
- Mary Catherine Laura for encouragement and financial support.
- Frank Swehosky III for help with debugging, program testing, technical support, and encouragement.
- Dr. Catherine Parsoneault for editorial and moral support.
- Timothy Bateman for program testing and making me laugh.
- Simon Jewell at SyGem Software for help with some elusive bugs.
- All the wonderful people at Treetops School International.

For encouragement, suggestions, and technical support:

- Brandon Wiley, Victor Lowther, Jonathan Leistiko, Malisa DiGiacomo, Austin Appleby, Nathan Walther, and many others.

TABLE OF CONTENTS

Page

LIST OF FIGURES

INTRODUCTION

In the book *The Child's Conception of Space*, Piaget and Inhelder (1956) argued that children develop spatial understanding in three stages. First, children comprehend topological relationships including proximity, separation, order and enclosure. Second comes an understanding of projective relationships such as perspective, proportions and distance. Third, children grasp Euclidean relationships involving measurement and coordinate systems. These later two stages occur in tandem, as the "concepts of projective and Euclidean space develop together and are mutually interdependent" (p. 419). Progressing through these stages of spatial development, the children move from holding an egocentric frame of reference, to seeing themselves as part of a coordinate system and being able to imagine other points of view within this system (Philleo, 1997, p. 22).

There has been debate about the number of distinct spatial abilities that exist. The general consensus posits that there are two—visualization and orientation. Spatial visualization tasks involve at least part of the imagined space being manipulated by moving or altering the objects. Spatial orientation problems require only changes in the viewer's perspective (Tartre, 1990, p. 216-217).

Furthermore, spatial abilities have been tested in relation to academic and career success. Frank Holliday found a strong correlation between spatial ability and performance in a mechanical drawing class and, later, success as engineers (Dixon,

---

This thesis follows the style and format of *Journal of Educational Psychology.*

1983, p. 47-48). A number of other studies have drawn similar conclusions with spatial ability being "linked primarily to success in math and science" (Philleo, 1997, p. 27). For example, Tartre (1990) found that "spatial orientation skill appears to be used in specific and identifiable ways in the solution of mathematics problems" (p. 227). These studies lead us to conclude that the development of spatial understanding should be a major concern of educators. Research on spatial ability benefits educators by providing better understanding of how spatial understanding can be coached successfully and can offer models for good practices. The resulting curriculum may in turn increase children's potential for success in certain academic subjects and careers.

Recent research has focused on using computers as teaching aides. For instance, in the 1980s researchers examined the benefits of teaching children the Logo programming language (Miller, Kelly, & Kelly, 1988). Mayer and Fay (1987) conceptualized a framework of three areas of cognitive growth that occur: learning the syntax of the Logo language, understanding the semantics of Logo, and transferring Logo knowledge to other realms. Research suggested that children can learn and transfer Logo concepts while bypassing the syntax and semantics via simplified versions of Logo (e.g., Watson, Lange, & Brinkley, 1992; Cohen & Geva, 1989; and Cohen, 1988). Most Logo research focused on the increased spatial abilities gained as a viable reason to teach Logo to elementary-age children.

If the most useful concepts gained from learning Logo relate to spatial ability, and if these concepts can be learned and related to other areas without learning the syntax and semantics of Logo, then why does Logo need to be the vehicle for increased spatial ability? Perhaps media that do not require syntax and semantics could fulfill the same purposes.

Some researchers have examined computer-assisted instruction (CAI) and its benefits (e.g., Miller, 1992, Baxter & Preece, 1999; and Gerson, Sorby, Wysocki, & Baartmans, 2001). With CAI, a computer is utilized to enhance instruction. The computer may be included as part of the teacher's presentation in the form of computer visuals, audio, or multimedia. Or the students may have the opportunity to personally interact with the computer. The use of CAI has been found to cause "significant learning increases, in less time, with more positive students attitudes toward the topics being taught" (McCuistion, 1991, p. 26).

Other researchers have focused on computer games as a tool for increasing spatial ability. Specifically, children with less well-developed spatial skills improved significantly and caught up with their peers by playing games that incorporated a large spatial problem-solving component (e.g., Subrahmanyam & Greenfield, 1994; and De Lisi & Wolford, 2003).

Many times what is most important in programs that aid spatial understanding is the child's ability to explore. Discovery occurs at the child's own pace in a manner that holds the child's attention because he or she is in control of the learning environment. For example, Philleo (1997) designed three-dimensional spaces called microworlds using Virtus Walkthrough Pro for children to investigate. He found that "microworlds can create powerful environments for engaging students in visual-spatial processes" (p. 81). Sachter (1991) took a different approach when she created a program in which children could place and transform objects in a 3D space, in addition to exploring an existing environment. In Sachter's program students not only learned about spatial concepts but also "actively and explicitly using [sic] them to construct images" (Sachter, 1991, p. 361). The children in her study improved their spatial skills "quantitatively and qualitatively regardless of their styles [of interaction with the program and instructor] and spatial ability" (Sachter, 1991, p. 361).

The development of exploratory and engaging software for children should be encouraged. These educational tools should:

- have age-appropriate content,
- be intelligently designed,
- be of high technical quality,
- be user friendly, and
- be easy for instructors to integrate into the curriculum.

A larger selection and wider availability of spatially oriented programs may lead to an increased number of children with better spatial understanding.

RESEARCH FOCUS

Rieser, Garing, and Young (1994) established that while children's "long-term spatial knowledge is functionally viewpoint independent," (p. 1271) the "spatial knowledge in working memory is organized so that it is functionally viewpoint dependent" (p. 1272). In other words, a spatial layout memory contains a combination of various perspectives, but once a particular vantage point has been brought to the forefront of a person’s thinking, extra processing must occur to change that perspective to another. In their experiments, Rieser, Garing, and Young asked the children to imagine a familiar environment (their classroom) from their seats. Once the children were oriented, the researchers asked the children to imagine the classroom from their teacher's perspective. The students tended to have quite a bit of difficulty doing so. However, if the researchers blindfolded the children and led them around the testing environment in a pattern similar to that in which they would walk to get from their seat in the classroom to the teacher's seat, the students could correctly orient themselves to the new perspective. Thus, the children were given a strategy for how to adopt a different perspective of a space once a particular viewpoint had been imagined. Walking through the space gave them a way to change their perspective.

Movement through physical space can be demonstrated in virtual computer environments by employing animation. Animation leads to a number of benefits in the realm of spatial understanding. Active movement is "the most effective means of gaining spatial information," and can be accomplished by physically moving through the space and manipulating the objects by hand, or by viewing this kind of movement and manipulation on film (Zavotka, 1987, p. 135). McCuistion (1991) reported that this type of dynamic film presentation increased mental rotation skills. Moreover, Zavotka's (1987) study showed that "viewing computer animated films of objects rotating and changing dimension" improves orthographic drawing interpretation skills (p. 143).

This project combines active movement through animation with a method of changing the point-of-view for a scene. As Rieser, Garing, and Young (1994) did, I gave children a strategy for how to adopt a different perspective of a scene once a particular viewpoint has been imagined. Instead of moving through a physical space, however, the children move through a virtual computer environment. This is accomplished with the motion of the camera in a 3D scene. As the camera moves from one position and rotation to another, the children see how the view changes. The virtual environment that the children explore with this changing point of view may be one of their own creation.

IMPLEMENTATION

In order to allow children to devise a virtual three-dimensional scene and move through it, I developed a computer program called Viewpoints. The program's graphical interface presents a 3D perspective view of an environment. Simple geometric objects such as boxes and spheres can be placed in the 3D space and modified by the child. The location, rotation, and size of each geometric form can be interactively manipulated to demonstrate how each characteristic relates to the object. Additionally, more cameras may be added to the scene. The positions and rotations of the cameras can be changed in the same manner as the objects' spatial properties.

When these characteristics are modified, the transformation from the old set of properties to the newly specified ones is shown through a sequence of images. These frames are generated in real-time to present the specific object and camera alterations chosen by the child. For example, if the child rotates a cube by 90 degrees, the frames show the cube rotating from its previous orientation to the new one.

In addition to demonstrating a method for how to adopt a different perspective for a space once a particular viewpoint has been imagined (spatial orientation), showing property modifications via a series of images helps students better understand how the graphical coordinates that they specify transform an object (spatial visualization). The graphical coordinates relate to the program's measurement system. Rotation, for example, is measured in degrees, while length is measured in the same units as width and height. Understanding how the measurement system correlates with an object is particularly important with rotation where the end result of a clockwise rotation of 90 degrees is exactly the same as a counter-clockwise rotation of 270 degrees. Visually demonstrating transformations may lead to an improved grasp of what modifying object properties does both in terms of numbers and functions.

Furthermore, I designed Viewpoints with its target audience of 9-12 year old children in mind. Piaget and Inhelder found that children of this age range are developing their understanding of projective and Euclidean relationships. Because this program is for children, including some who may be relatively inexperienced with this particular kind of computer interaction, Viewpoints must have an intuitive interface and be easy to access. This should allow the children to focus on the spatial orientation and spatial visualization aspects without the distraction of an excessively complex interface or complicated installation process.

PORTABILITY

For this project, portability is characterized as the ease with which the target audience of students and teachers can obtain and use Viewpoints. Portability considerations influenced the choice of delivery method, programming language, and programming library selections.

For Viewpoints to best serve its role as educational software, it needed to be widely available and free. The most practical way to accomplish this was to make the program available on-line, so that anyone with an Internet connection could use the program. Schools could supply Viewpoints to students on any number of machines and students who need or would like extra time with the program outside of school could use it at their homes.

Because the program could be accessed from any computer, it is not possible to predict the operating system (Windows or Mac for example) of the computer upon which the program might be run. Consequently, the program could not contain code that only works on a specific operating system; in other words, no native code. Java seemed to offer the ideal choice in language, because of its multi-platform capabilities and Internet compatibility.

Selecting Java as the programming language allows a unique opportunity in delivery method. A utility called Java Web Start (JWS) exists to simplify the download process for Java applications. JWS keeps track of the files needed for the software to run and downloads them automatically. Then it stores these files locally for future use. Thus, unlike many traditional Java programs available on-line, once the applications have been accessed through JWS they can be run even when there is no Internet connection. Furthermore, if the computer in use does re-establish an Internet connection, JWS automatically checks for an updated version of the software.

Another valuable aspect of JWS is its built-in security. JWS encloses the software within a secure environment. As a result an application cannot access the local computer's file system without the express permission of the user. In other words, JWS prevents programs run in its environment from installing malicious code, for example, computer viruses, on the user's computer.

Next, I had to select appropriate programming libraries. A programming library is a collection of existing code, including classes and methods, designed to fulfill a certain purpose. For Viewpoints, I utilized two libraries: one to help build the graphical user interface and one to aid in the construction of the three-dimensional virtual environment. For each of these libraries, the same restriction applied as for the language selection—they cannot contain native code. Also, they must be written in Java.

For the interface, I selected Java Swing. This library consists of a compilation of classes each supplying a basic building block of a graphical user interface. These elements are highly customizable and multi-purpose. I choose the Swing libraries because they provide for comprehensive and flexible graphical user interface development.

Choosing an appropriate library for the construction of the 3D environment required more deliberation. I narrowed the options to two libraries: Java 3D and Jazz3D. These supply similar functionality: a graphical presentation of a 3D environment; the creation of objects within that environment; and the ability to view those objects from a specified vantage point.

The inclusion of native code in Java 3D is the main difference between it and Jazz3D, which contains none. Native code benefits Java 3D by providing faster processing and faster image rendering because it utilizes specialized hardware-specific code. The inclusion of this code means that a program using Java 3D needs a different version of the library depending on the user's operating system. On the other hand, Jazz3D is consistent across platforms.

Native code also complicates the download process in another way because JWS requires special permission from the user to download any file with native code in it. If

JWS does not need to ask for any special security permissions then it can keep the program completely within a secure environment. By doing so, the administrator of a computer classroom knows the software will not jeopardize the school's computer security.

Because Jazz3D does not have the added complication of native code, I selected Jazz3D over Java 3D. However, Jazz3D does present its own set of hurdles. First, in order to fully integrate Jazz3D with the program Viewpoints, I had to obtain a full source code license for Jazz3D. This allowed me to customize Jazz3D to facilitate successful interaction between it and Viewpoints. As part of this process I added many new functions to Jazz3D. One of the complications I dealt with was that Jazz3D only provides support for one camera while Viewpoints needs to utilize any number of cameras. Consequently I had to develop and implement a schema for storing and manipulating a large number of cameras. Additionally, I created complex animation algorithms for object and camera motion.

Combining Java, Java Web Start, Swing libraries, and Jazz3D provides a platform-independent program that can be made available on-line via a download manager. Thus the resulting product can be easily downloaded and accessed from any computer with an Internet connection.

GRAPHICAL USER INTERFACE

To allow the program's user to focus on spatial orientation and spatial visualization, the graphical user interface (GUI) must be intuitive and simple enough to

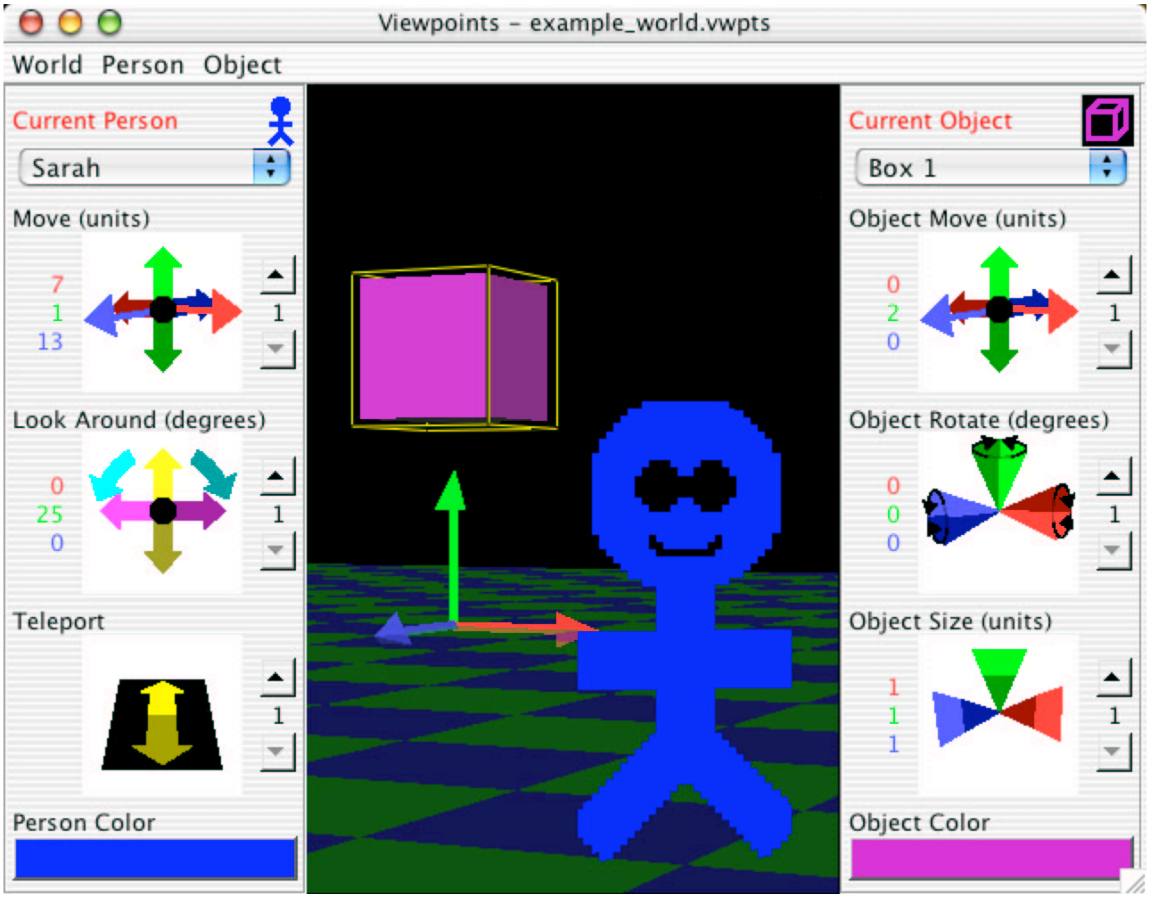

Figure 1. Viewpoints' graphical user interface.

understand with very little introduction. As Metros and Hedberg (2002) point out, "effective GUI design will ensure that the learner's focus [is] on learning rather than operating the software" (p. 204). Therefore, Viewpoints is mouse-driven, with as much information presented pictorially as possible (see Figure 1). A visual relationship exists between each interface element in which a property can be manipulated and the result of that manipulation.

Equally important, the program needs to relate to the child's existing frame of reference. Metros and Hedberg (2002) point out that using visual metaphors "reinforces consistency, takes advantage of previously learned associations, promotes understanding between diverse concepts and helps the learner grasp complex ideas" (p. 203). To aid in this, each camera in Viewpoints is referred to as a *Person* and is represented by a stick figure in the 3D environment. This equips the child with a symbolic representation of him- or herself in the virtual environment, otherwise known as an avatar. Preparing a child for a viewpoint change by providing an avatar for the new viewpoint "reduces or even abolishes viewpoint dependence for detecting a change in an object location" (Amorim, 2003, p. 158).

Similarly, it is important to establish a visual frame of reference within the 3D environment. Amorim (2003) found that supplying avatars and an architectural environment improved spatial judgments more than supplying avatars alone does (p. 191). In Viewpoints, coordinate axes, a ground plane and a default static camera furnish non-movable environmental elements. Colored arrows extending from the center of Viewpoints' coordinate system depict the X-, Y- and Z- axis directions and convey not only the location of the center within the system, but also its orientation in relation to the current view. The semi-transparent ground plane denotes an additional visual guide for "down" and the checkered pattern of one-unit squares on the ground plane helps clarify the relative size of one unit in Viewpoints' coordinate system. The default camera provides another way to orient the child's view of the 3D environment. This camera always exists and cannot be changed. It supplies a stable view of the world to which the child can always return.

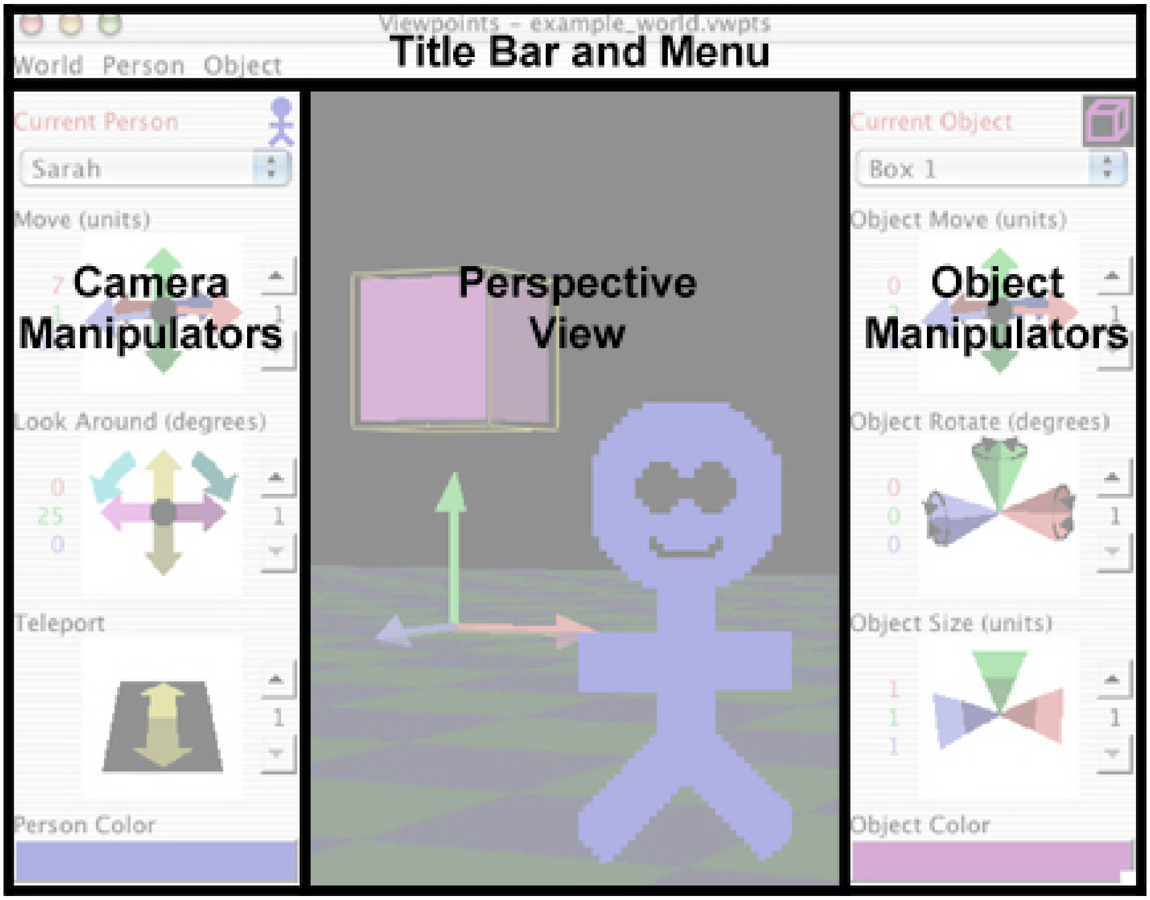

Figure 2. General layout of Viewpoints' interface.

Overall, the GUI consists of three main components to organize the data being illustrated (as shown in Figure 2). The center presents a perspective view of the 3D environment. The left side provides areas to manipulate the properties of this particular camera. The right side allows for modifications to the currently selected object.

The original idea for the interface also included two-dimensional orthographic views, but these added an unnecessary layer of visual complexity that could became confusing to the viewer. To further simplify the interface, Viewpoints has pull-down menus for functions that do not lend themselves well to being displayed all the time, such as adding and deleting objects and cameras.

Viewpoints should be internally consistent to “[allow] a user to transfer knowledge and experience acquired in using other components to new tasks” (Hartley, 1998, p. 7). I selected descriptor words with the target audience of 9-12 year olds in mind and then used them where appropriate. On a more comprehensive level, I designed the layout and behavior of graphical elements carefully, taking special notice of balance and coherency. Components used for camera modifications visually align with their counterparts used for object manipulations. Functions with similar purposes act and look similar. When appropriate, menu options matched in word choice and purpose to standard menu options available in other computer programs.

Consistent use of color plays an important role in revealing connections between the 3D environment, the cameras, the objects, and the graphical user interface elements that provide the ability to manipulate camera and object properties. Each axis in the environment (X, Y, and Z) is assigned a particular color. Every interface element involving that axis displays its correlation to that axis by using the assigned color. For example, the X-axis is red. Any numerical presentation of the measurement along the X-axis is red as is any polygon in the GUI representing that axis. To further demonstrate the connection, the X-, Y-, and Z-axes that extend from the origin of the 3D environment also match the assigned X, Y, and Z colors. Throughout Viewpoints’ interface, each use of a specific color visually links that element with its corresponding arrow in the 3D environment.

THE WORLD

For this project, the term *world* refers to an arrangement of cameras and objects in the 3D environment. The center window of Viewpoints' interface displays the world as seen from a specific camera.

As part of the program's mouse-driven capabilities, the objects and cameras that are visible in the perspective view can be selected with a mouse click. When an object is selected, that object can be manipulated. When a camera is selected, the view changes frame-by-frame to become the view seen from the selected camera. After this view change, the newly-selected camera can be manipulated.

Additionally, open and save capabilities are available. The saved file holds information about the current state of the 3D environment including all the properties of each camera and object in the world. Open and save capabilities allow teachers to save worlds for future use in classroom assignments or demonstrations, and also provide children a way to preserve their work as part of an assignment or as a creative endeavor. The drop-down menu entitled *World* supplies this functionality. This menu also offers the opportunity to open existing worlds. Furthermore, this menu provides the option to hide or show the ground plane.

CAMERAS

As mentioned, in order to affirm the relationship between the child's point of view of the world and the camera's point of view, each camera in Viewpoints is referred to as a Person and is represented by a stick figure. Each stick figure illustrates the properties of its corresponding camera by matching its location, rotation, and color to that of the camera. The point in the 3D space where, mathematically speaking, the camera resides aligns with the eyes of the stick figure. In addition, each Person can be given a name, thereby adding another level of relationship to the child's existing frame of reference. For example, the child can go to see what the Person named Sarah is looking at and the 3D view will change to Sarah's point of view, out of the stick figure's eyes, via animation.

General camera functions are available in the *Person* drop-down menu next to the *World* menu. From here cameras can be added to the world using *New*, which places the new camera in the same position and rotation of the default camera. Or an existing Person can be copied, which adds a new camera at the same location and orientation as that Person. Cameras may also be deleted, with the default camera being the exception. The *Person* menu also gives the user the opportunity to rename any camera except the default camera.

Particularly important in this menu are the *Go To* and *Look At* functions. *Go To* discloses relationships between cameras. When a specific Person is selected as the target of *Go To*, the view animates from the current camera's point of view (see Figure 3) to the point of view of the target camera (see Figure 4). In the process, the child sees how to move and rotate through space to gain the perspective of the destination Person. Likewise, the *Look At* function shows animation between two points of view. However, instead of moving between two different cameras, *Look At* animates the change between two orientations of the same camera. With *Look At* the child can select any other camera

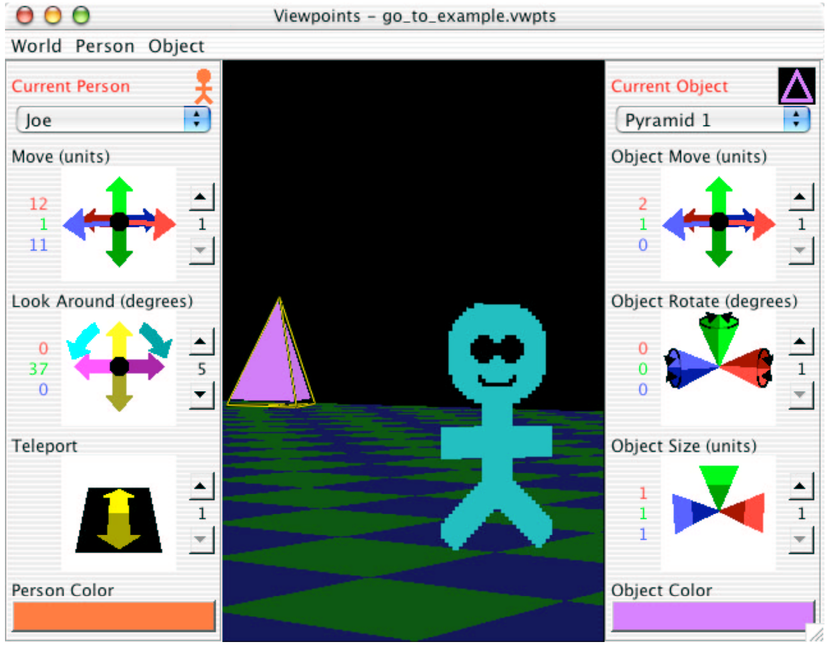

Figure 3. Screenshot before example *Go To* action.
World as seen from Joe's point of view. Kelly (in cyan) selected as the target of *Go To*.

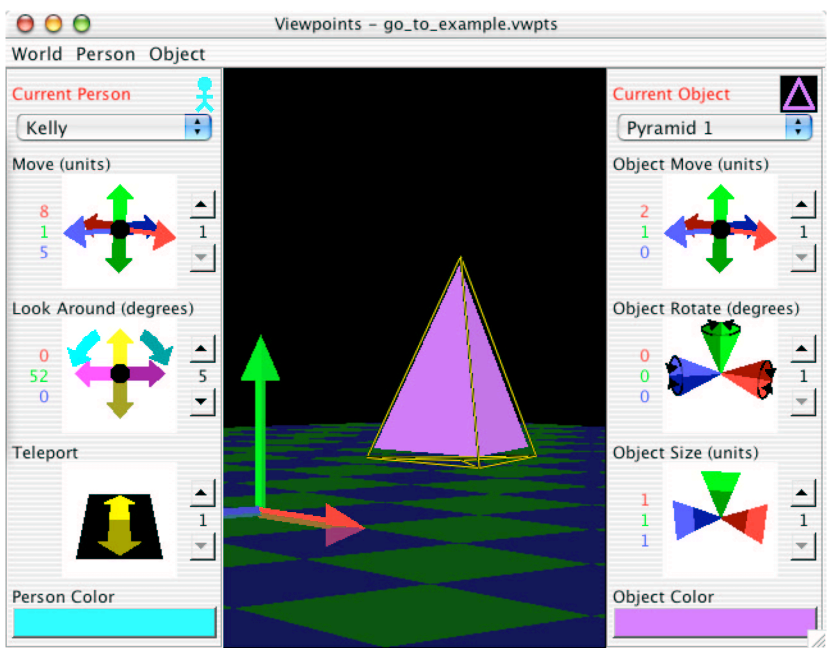

Figure 4. Screenshot after example *Go To* action.
World as seen from Kelly's point of view.

or object in the world to view. The current camera then animates the change to its new orientation, pointing at the selected object or camera.

That is not, however, the only way to modify the current Person. As mentioned previously, the left side of Viewpoints' interface provides the opportunity to modify various characteristics of the current camera including its location, orientation, and color. To help clarify what can be currently modified, a small stick figure icon in the color of the current camera appears on the top right side of this interface component. A drop-down menu within this component supplies an additional way to select a camera. Selecting a different camera triggers the same action as the *Go To* menu item; that is, an animation showing the current view moving and rotating to assume the newly-selected camera's position and rotation.

The position, rotation, and teleport manipulators each show the same basic layout consisting of three parts. The left part gives the numerical values for x, y, and z for that property shown in colors coordinated with their appropriate axes. The middle section presents a pictorial representation of how that property changes. Here the depiction of each axis can be clicked on to modify the value of the property along that axis. When the mouse moves over an axis that can be changed, a highlight appears around that axis. The right section offers an increment counter to modify the amount the value changes when an axis is clicked on.

The location modifier, titled *Move*, shows arrows to represent the directions the camera can move. Clicking on an arrow moves the camera in the corresponding direction. These arrows automatically align with the world coordinate arrows as seen

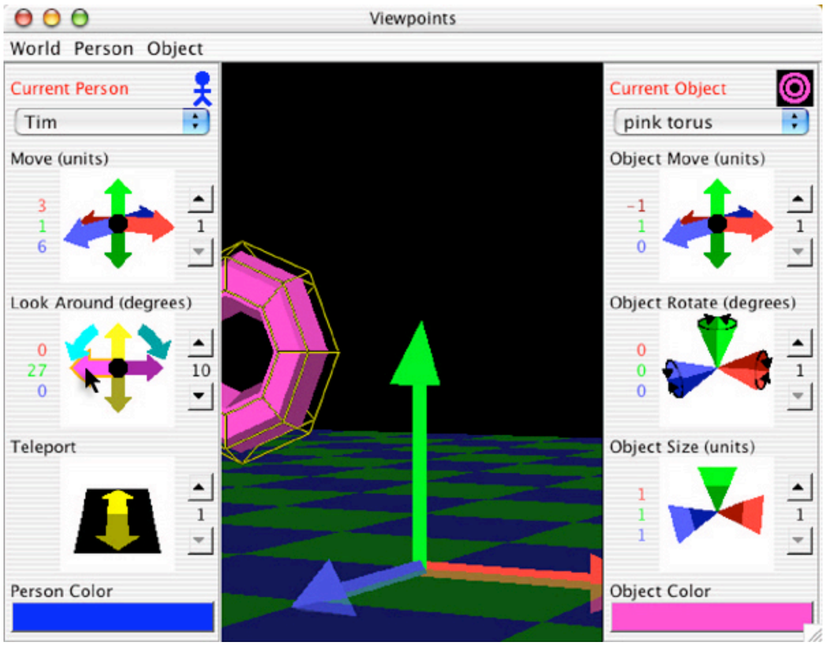

Figure 5. Screenshot before changing the rotation of a camera.
Note the cursor is at the pink arrow in the *Look Around* property manipulator.

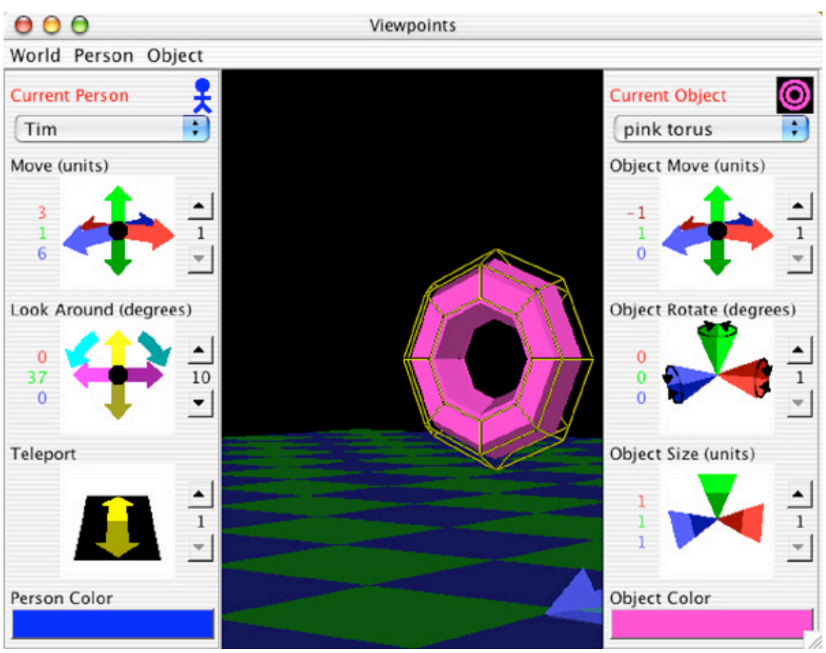

Figure 6. Screenshot after changing the rotation of a camera.
The current Person (named "Tim") is now looking further left than previously.

from the current camera. These arrows are also color coordinated so that the red arrow is parallel to the red arrow in the perspective view, the X-axis. For movement in the negative direction along each axis, an arrow pointing in the opposite direction is provided in a darker shade of the coordinating color, dark red for movement in the negative X direction, for example. If due to the camera's current position and rotation, two arrows nearly align, the arrow in front becomes narrower so that the arrow behind can still be selected. Also, some arrows have perspective to clarify that they seem to be going into or out of the screen.

The rotation modifier, titled *Look Around*, also presents arrows, but with a different purpose and arrangement. Clicking on one of these arrows changes where the camera is pointing (see Figures 5 and 6). These arrows are different colors than those for position because these coordinate with the camera's pitch, yaw, and roll, not the world axes.

The next modifier, titled *Teleport*, moves the camera forward and backward along the camera's local Z-axis. Essentially it provides another method of moving the camera, similar to a dolly move on a movie set. This is illustrated by a set of arrows on a plane in perspective such that they appear to be going into and out of the screen. Because this move may not line up with a specific world axis, these arrows are not color coordinated with any of the world axes.

Lastly, the bottom button provides a way to select a different color for this Person. With the Person selected, the user clicks in the color button and chooses the desired color from the color palette that appears. Not only does this allow

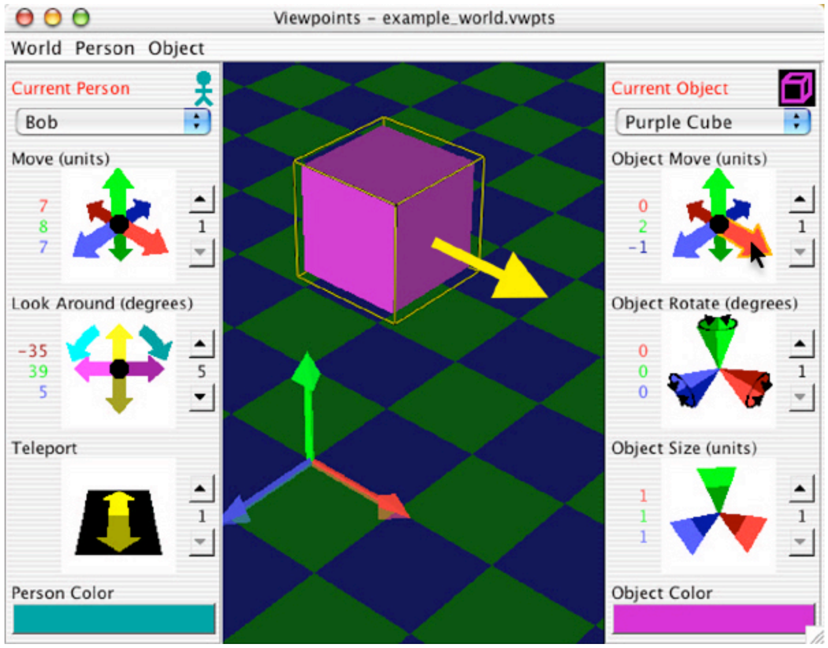

Figure 7. Screenshot before changing the position of an object.
The large yellow arrow by the purple cube (not included in Viewpoints) shows the direction the cube will move when the red *Object Move* arrow is clicked.

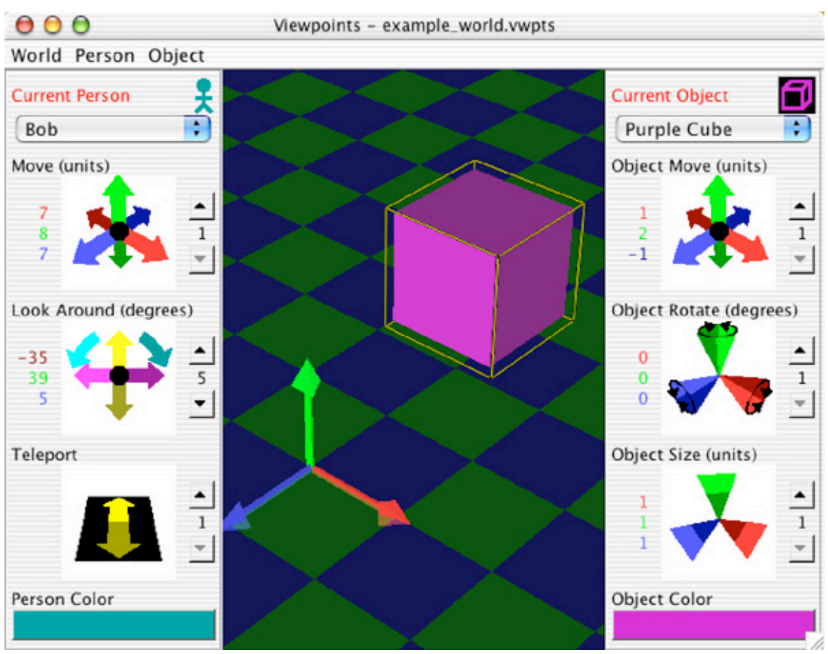

Figure 8. Screenshot after changing the position of an object.

personalization of the cameras, but it also helps the user distinguish the cameras within the 3D environment.

OBJECTS

Like cameras, objects can also be added to the world and modified. Accordingly, general object functions can be found in the *Object* drop-down menu next to the *Person* menu. From here new objects can be created and existing objects can be selected, renamed, copied, and deleted. Object types available are cube, cylinder, pyramid, sphere, and torus.

As mentioned, the current (selected) object can be modified using the right side of Viewpoints' interface. A small icon representing the current object's type, and shown in the current object color, depicts what can be changed. A drop-down menu exists to select a different object. In the perspective view, a wire-frame encompasses the current object to signify it.

For objects, the position modifier is titled *Object Move* to help distinguish it from the camera's location manipulator. *Object Move* appears and functions, however, the same way. Clicking on an arrow moves an object in that direction along that axis (see Figures 7 and 8).

In contrast to the location arrows, color-coordinated cones show how rotation occurs around an axis, not along it. Each cone is divided down the center with one side for clockwise rotation and the other for counter-clockwise rotation. The rotation modifier is called *Object Rotate*.

Objects have an additional characteristic that can be changed, scale. *Object Size* is portrayed as triangles extending out along axes. The outer part of a triangle appears to be expanding and thus symbolizes positive changes in size along that axis. The inner section of a triangle, however, appears to be shrinking and therefore signifies negative changes in size along an axis. See Figures 9 and 10 for an example of scale change.

The color of an object can be modified in a similar manner as the color of a camera. The *Object Color* button is positioned on the right side of the window.

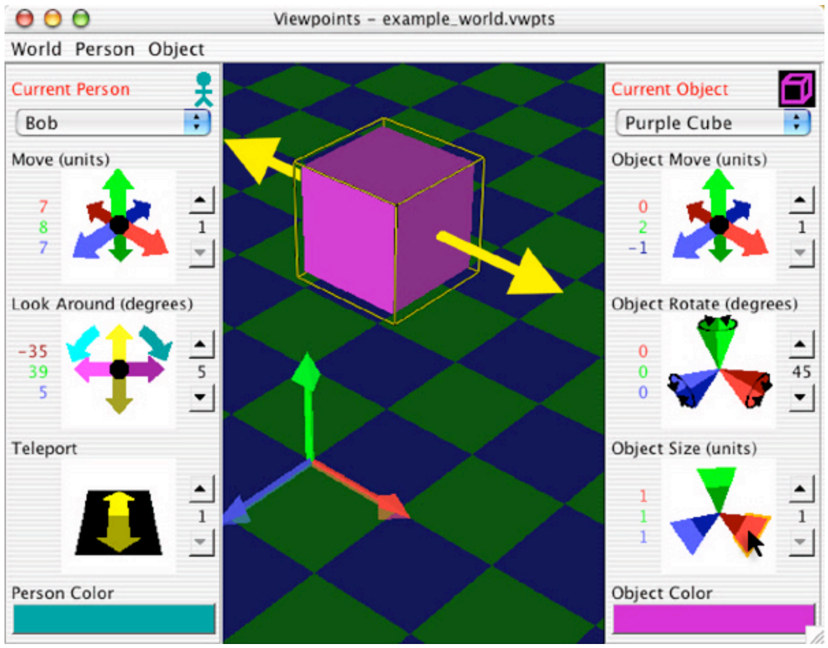

Figure 9. Screenshot before changing the scale of an object.
The large yellow arrows (not included in Viewpoints) demonstrate the directions the cube will expand when the red *Object Size* polygon is clicked.

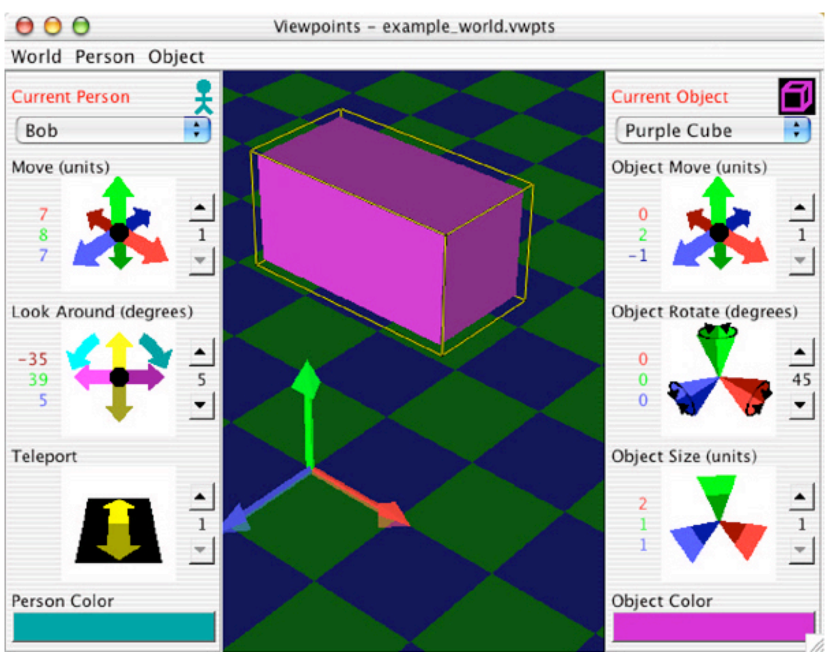

Figure 10. Screenshot after changing the scale of an object.

METHODOLOGY

I designed a limited study to evaluate Viewpoints' usefulness as an educational tool in terms of content, instructional design, technical quality, student use, and instructor use. Because Viewpoints is based on past research in which improvements in spatial skills were found, I believe Viewpoints would also do this. However, scientific proof will be left to future work. This study took place in the Summer of 2004 in a small charter school in a suburban area of the Southwest United States.

Participants included both students and teachers who met in the computer lab of their school for the study. I gave a brief demonstration of Viewpoints to all volunteers. The students and instructors then paired up at various computers in the lab. Each child received an activity booklet to help guide them through how to use the program and how to do various spatial visualization and spatial orientation tasks. The educators observed their students interacting with Viewpoints during this stage, and then had the opportunity to work with the software themselves. Finally, the instructors each filled out an evaluation form.

In addition to important feedback regarding the program, this study also assists educators by providing a learning tool with application beyond the classroom. Introducing this and similar programs into school curricula may assist students to develop spatial abilities they may draw upon later. Benefits for the children in the study may include improved spatial orientation and visualization. Both these spatial skills should help these children in their future schoolwork—particularly in math, science, and art.

PARTICIPANTS

Because Viewpoints has a definite target audience within an educational environment, teachers, and students can offer valuable insights into how well the program can be utilized in that setting. Participants consisted of student and teacher volunteers from a small charter school in a suburban area. No payment was bestowed, but all participants received the web site location of the program in case they wished to work with it in the future. This research study was reviewed and approved by the Institutional Review Board – Human Subjects in Research, Texas A&M University.

As this study is designed to gather general feedback about the software with an eye towards its use as an educational tool, a small number of participants were deemed sufficient and no separate control group was considered necessary.

The six student volunteers (two girls and four boys) were from 9 to 12 years old. All had prior computer experience including drawing pictures, playing games, and searching the web. Other areas of computer experience varied. Most had written papers or stories, and worked with educational programs. Some had also chatted on-line, used email, or made their own webpage.

The five volunteer instructors, four women and one man, had taught a wide variety of subjects, singularly and as a group, including but not limited to, language classes such as English and Latin, science, history, physical education, math, social studies, and psychology. Grade levels taught either currently or in the past ranged from elementary through the community college level. Years of experience as an educator ranged from 3 to 25. Experience with computers also varied widely from very limited up to 13 years. The teachers' familiarity with computers was in a broad number of areas including, but not limited to, graphics, web related activities, word processing, and programming.

APPARATUS

Because a classroom setting is one of the intended environments for using Viewpoints, the study was conducted in the computer lab of a small suburban charter school. The volunteers all attend or teach at this school. This provided a familiar environment, as would be the case in the program's intended home or school use. The computers all ran the Windows XP operating system and had access to the Internet. Before the participants entered the lab, I downloaded Viewpoints to every computer from the program's website, so each child could select any computer in the lab.

I provided each student with an activity booklet to use as a guide for learning the program. The booklet presents a series of increasingly complex activities designed to include both spatial visualization (object transformation) and spatial orientation (camera modification) tasks. Instructors also received copies of the activity booklet to follow along.

Additionally, I gave each teacher a copy of the Educational Software Evaluation System (ESE System) by Arnie Abrams (1995, p. 65-68). The ESE System supplies a way for educators to rate software in terms of content, instructional design, technical quality, student use, and instructor use. As a checklist-style assessment tool, the ESE System provides a method to review the software on a formative level, appraising the technical quality, usability, and "appropriateness of software design features" (Tergan, 1998, p. 17). This stage of information gathering is appropriate to a newly developed program like Viewpoints to "[obtain] data to increase the effectiveness of [its] design" (Shiratuddin & Landoni, 2002, p. 175). A later study, not within the scope of this project, could then take a summative approach to gather empirical evidence to determine if the program does increase spatial knowledge. This later type of data collection would occur after appropriate measures had been taken to enhance Viewpoints based on the information gathered in this current study. In other words, the ESE System provides Viewpoints' first level of review.

Because "the reliability and validity of criteria in developing a software evaluation instrument has seldom been the subject of empirical analysis" (Tergan, 1998, p.11), the reliability and validity of the criteria used in the ESE System remains unknown. Tergan (1998) concluded that one-dimensional criteria in checklist-style evaluation forms "are useful and quite unproblematic," whereas two-dimensional criteria present a more complicated issue that may limit the criteria's relevance (p. 12). A criterion is inherently two-dimensional if the rating of that criterion depends on both a feature of the software and the cognitive preconditions of the student using the software (Tergan, 1998, p. 12). Consequently, while most of the criteria in the ESE System are one-dimensional, the criteria involving student interaction with Viewpoints in a manner that depends on the characteristics of the student, must be evaluated within the context of those who participated in this study.

PROCEDURE

After I collected the consent forms from the parents and teachers, I verbally explained the consent form to the children. Each of the six children chose to participate and sign the consent form. I handed out the ESE System to the instructors so they could become familiar with its criteria and rating system. The students and educators filled out their respective Biographical Data forms.

Next, I presented a brief demonstration of the software program and answered a few questions. The students and teachers paired up. Because six students and five instructors participated, two teachers volunteered to jointly observe the extra student. After opening the program on each computer, the children and educators received copies of the activity booklet to act as guide for the program. The activity booklet (see Appendix B) included activities that asked the children to change the size, location, and rotation of various geometric objects in the 3D virtual space. The booklet also included the task of viewing the scene from different vantage points as designated by small stick figures in the space. Some of the children were so excited about the creative opportunities presented by Viewpoints that they skimmed the activity booklet and then jumped right into working on their own creations. While developing their own worlds in the 3D virtual space, these students created and modified objects and viewpoints just as they would have had they been doing the activities in the booklet.

One student did choose to stop interacting with the program earlier than the other children. I had been told in advance that this would probably occur because this particular child has autism. The student's mother attended the study and determined when he should stop.

After the teachers observed the students' interactions with the program, most of them chose to interact with the software also. Each teacher then each completed the Educational Software Evaluation System form handed out earlier. After collecting the evaluation forms, I gave all participants the web site address for Viewpoints so that they could access it later if they wished. Additionally, many of the students and educators asked to keep the activity booklet, which I gave them permission to do. I also supplied each parent, teacher, and student with a copy of the consent form he or she signed.

RESULTS

The Education Software Evaluation System provides a way for teachers to evaluate whether a computer program is an appropriate educational tool for their needs. The form consists of 30 specific areas to score along with open-ended questions for more general thoughts and comments. The scored indicators are divided into five categories: content, instructional design, technical quality, student use, and instructor use. Each score is on a 0 - 100 scale. Guidelines provided by the ESE System for numerical rating are:

80-100 = excellent

60-80 = good

40-60 = average

20-40 = below average

0-20 = poor

Due to the small nature of the study all responses for a particular question are shown in the graph. It should be noted that one of the teachers chose to write comments instead of numerical evaluations. Thus, the graphical interpretations of data only illustrate up to four ratings for each scored area. The criteria in the figures are presented exactly as they are worded in the ESE System.* A summary of comments from all educators is provided in Appendix A.

---

* From Arnie Abrams *Multimedia Magic: Exploring The Power Of Multimedia Production.* Published by Allyn and Bacon, Boston, MA.

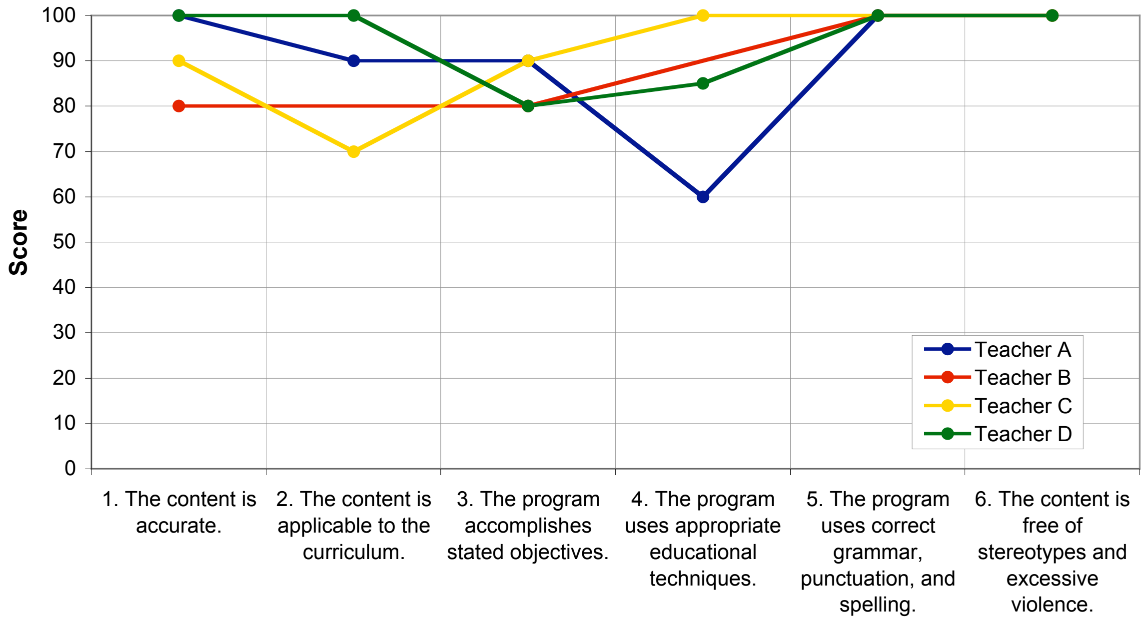

Figure 11. Content as rated by participants.

CONTENT

The ESE System provides criteria to rate content by its accuracy and applicability, achievement of stated objectives, use of appropriate educational techniques, use of proper English, lack of stereotypes and violence (see Figure 11). Participants ranked Viewpoints very high in use of proper English and lack of stereotypes and violence. Ratings were primarily in the excellent range (80-100). One teacher rated applicability to the curriculum 70 and a different teacher rated use of educational techniques 60. The addition of supplemental materials may be appropriate to address some of the concerns raised by this variability.

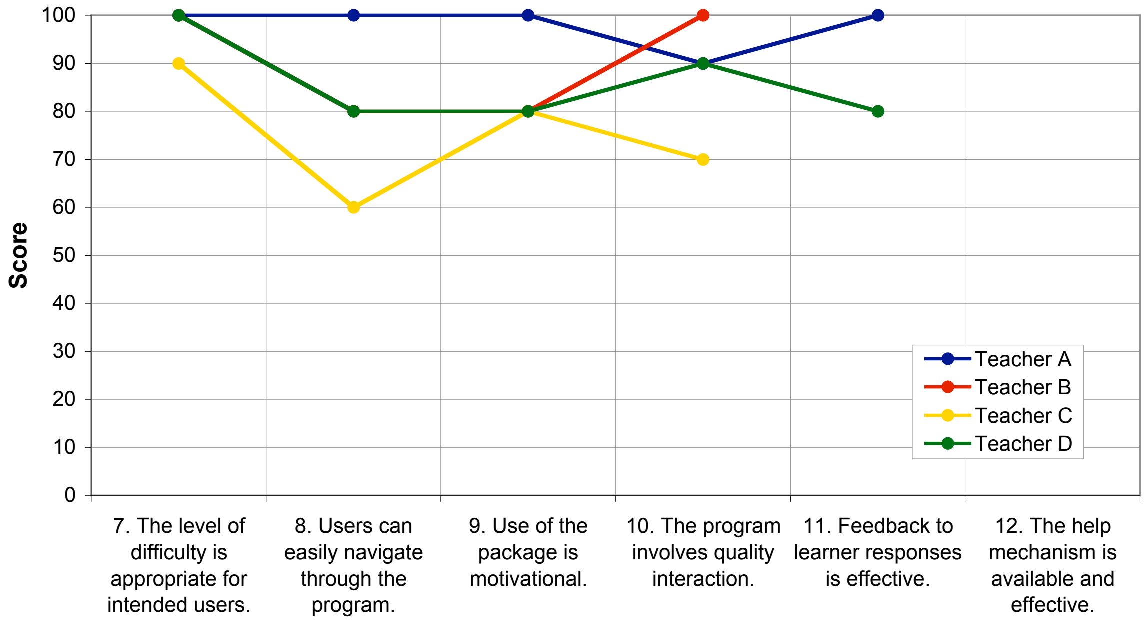

Figure 12. Instructional design as rated by participants.

INSTRUCTIONAL DESIGN

The difficulty level, ease of navigation, ability to motivate, quality of interaction, and effectiveness of the program's feedback and help mechanisms characterize instructional design (see Figure 12). Criteria 7, 8, 9, and 11 involve the cognitive preconditions of the students in the study and, consequently, reflect the measures' two-dimensionality. Most scores are in the excellent range (80 to 100) with slightly lower scores appearing in navigation (criterion 8) and quality of interaction (criterion 10). Teachers' comments mirrored the variation in navigation scores: "[Student] appeared to go through the program effortlessly" and "[Lacks a] click and drag component." Criterion 12 was not rated since Viewpoints does not have a help mechanism.

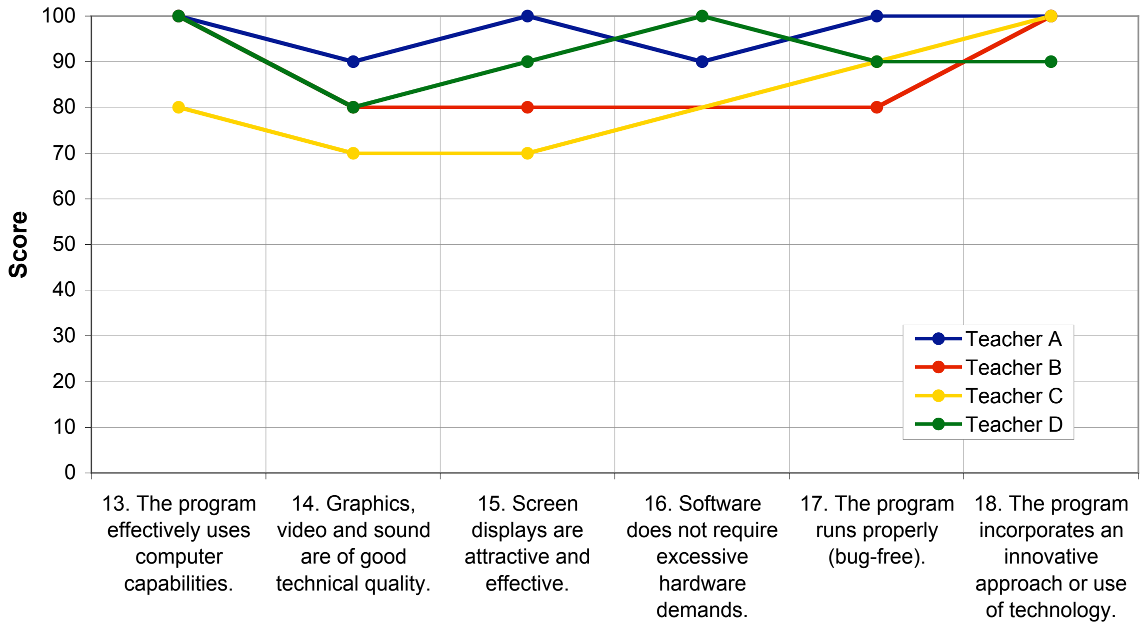

Figure 13. Technical quality (program operation) as rated by participants.

TECHNICAL QUALITY

Technical quality relates to the demands the software places on the computer (criteria 13 and 16), the visual quality of the program (criteria 14 and 15), the software's bug-free status, and whether the program incorporates an innovative approach. As illustrated in Figure 13, most scores portraying technical quality are in the excellent range (80 to 100). The scores for visual quality varied more, with one score of 70 in each category. These may be related to one of the teacher's comments, "students may become frustrated with speed," because that could be interpreted as a video quality issue or as component of an effective display.

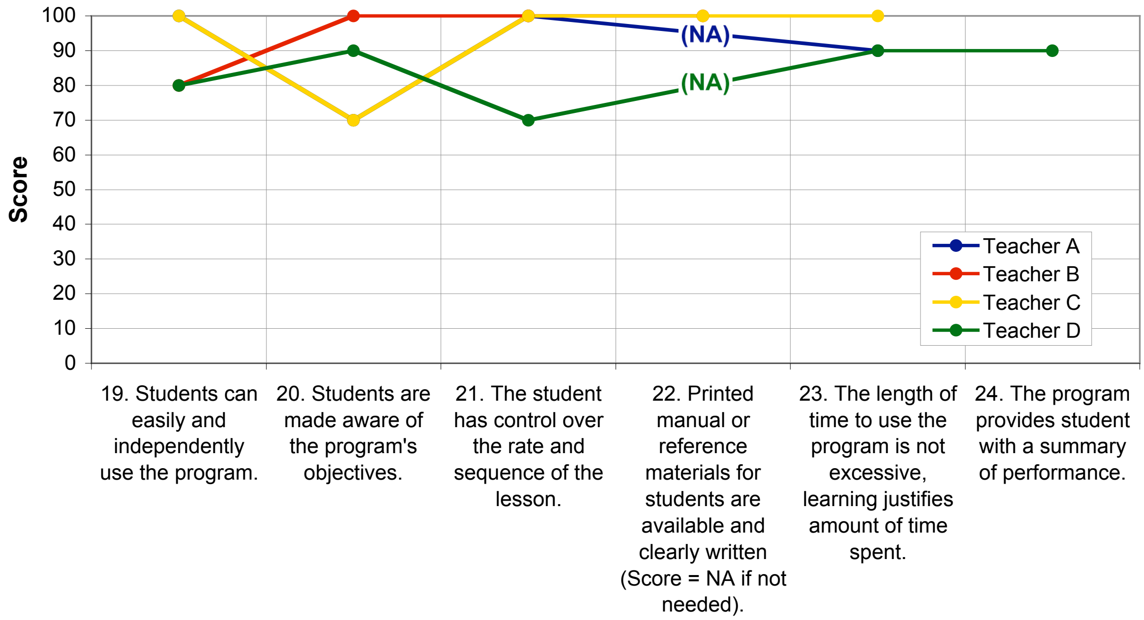

Figure 14. Student use (user friendliness) as rated by participants.

STUDENT USE

The ESE System categorizes user friendliness as independent use of the program, awareness of the program's objectives, control of the lesson speed, clarity of reference materials, length of time using the program, and the summary of performance (see Figure 14). Most scores were in the excellent range (80 to 100) with a larger variability in scores for criteria 20 and 21. Because ease and independent use of the program depend on the computer literacy level of the students, ratings for criterion 19 reflect the two-dimensionality of this measure. The two 100 scores for criterion 22 may reference the activity booklet provided as part of the study. Criterion 24 was not rated because Viewpoints does not contain explicit lessons, nor, therefore, a summary of performance.

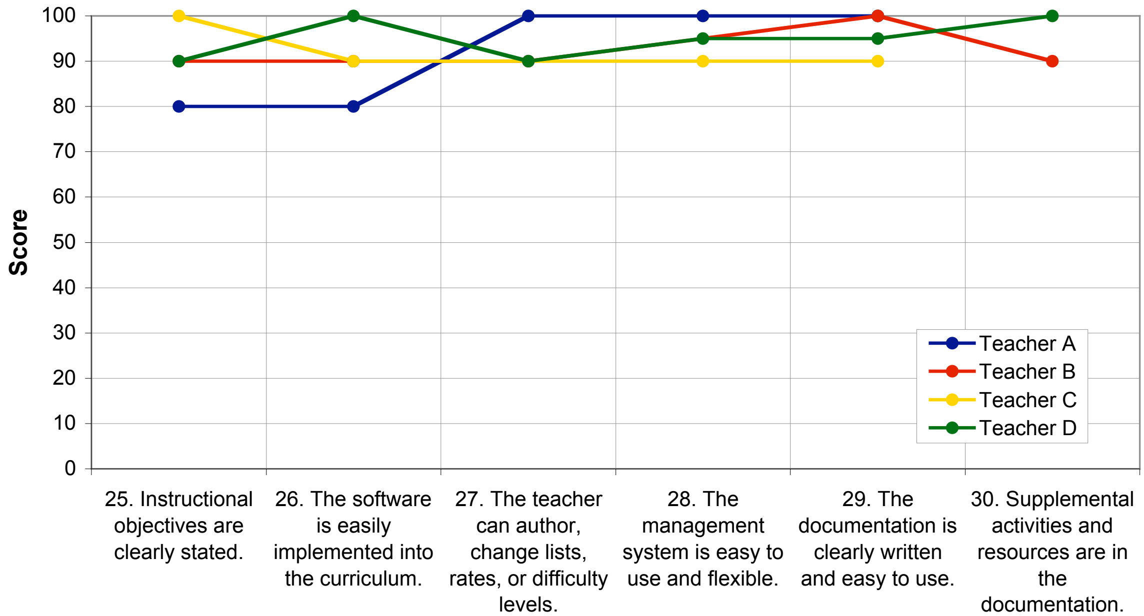

Figure 15. Instructor use as rated by participants.

INSTRUCTOR USE

Instructor utility is illustrated by the ease with which teachers can integrate the program into the curriculum (criteria 25, 26, and 30), and the ease with which the teachers can change the software to fit their needs (criteria 27, 28, and 29). Scores in all of these areas were in the excellent range (80 to 100) suggesting the program reached this goal (see Figure 15). These rankings can be further qualified by some of the instructors' comments. One educator opined that Viewpoints "can be useful used in a supplementary method in a classroom." Another felt that "many applications are evident." The teachers mentioned potential uses in math, science, language, geometry, and fine arts. For example, one instructor suggested that Viewpoints could be utilized to model our solar system, which is a complex concept to teach. The model could present the sun, planets, and moons scaled appropriately to show their relative sizes and distances. Additionally, all the elements could rotate in accordance to both their local axes and their orbits.

VANTAGE POINT CHANGES

As a supplemental question to the ESE System, educators were asked if they thought Viewpoints helps students better understand changes in point of view. Of the four who answered, all responded with “Yes”. Other written comments correlated with this, including a recommendation to obtain the software for spatial need. Some descriptions of the best aspects of the software were: “different viewpoints of 3-D software;” “manipulation of figures;” and “it allows students to actually see things that lie in a plane.”

One of the students who participated in the study has autism. Although he stopped working with the program earlier than the other students, his mother, who was also present, mentioned that he had worked with Viewpoints longer than she thought he would. She also opined that understanding changes in points of view better is a “huge consideration for an autistic child.”

PORTABILITY AND USABILITY

While the ESE System does not directly address the question of portability, some secondary evidence can illustrate the effectiveness of the measures taken to ensure platform independence. It should be noted that I developed Viewpoints entirely on a Mac OS platform in a home setting, while in the study Viewpoints ran in a Windows XP environment in a school setting. Information can also be gathered from the process I used to install the software on all the computers in the lab before the study began. The procedure conducted at each computer consisted of setting a web browser to the appropriate web page, clicking the Start button, and then selecting OK at the next prompt. No further action was needed on my part. Java Web Start determined what files to download, where to download them from, downloaded the files, and automatically started Viewpoints.

In terms of usability, the scores in instructional design, student use, and instructor use hint that Viewpoints provides a usable interface. Adding supplemental materials including reference manuals for students and teachers, suggested curriculum uses, lessons, and a help mechanism could improve Viewpoints' usability.

Within the limited range of this study, these results suggest that Viewpoints has strong potential to be an appropriate educational tool. Further development based on the improvements suggested in this study, followed by an empirical study of its effectiveness in increasing spatial understanding, could be done to confirm these findings.

CONCLUSION

The end result of this research and development is a Java computer program designed to help children understand a strategy for changing vantage points within an imagined space, and gain knowledge of how spatial transformations work. The limited evaluation process reviewed the software in terms of content, instructional design, technical quality, student use, and instructor use. This assessment provided valuable information on ways to enhance Viewpoints. The software's focus on portability and graphical user interface design should make learning spatial visualization and orientation easily accessible to anyone with an Internet connection.

In many ways, Viewpoints is similar to Sachter's (1991) program J3D, with the addition of demonstrations of object transformations through sequences of images. Viewpoints also incorporates Philleo's (1997) discovery that children like to manipulate objects in 3D computer environments (p. 80). Several children participating in the study illustrated this by declaring "Cool!" during the study. Five took the opportunity to create their own 3D worlds and appeared to enjoy themselves. Participating educators also expressed enthusiasm about the software's potential. Many had ideas of how Viewpoints could be incorporated into their curriculum and how it could be used to show difficult concepts.

Additionally, Viewpoints provides a basis for future work. Changes and additions could be made and studied. In particular, issues regarding the speed of the program could be addressed. Specifically, what determines the speed should be examined. Speculatively, Viewpoints may be perceived as slow because it animates any changes that occur, even those that are not directly visible. Adding a status bar to inform the user of any event currently processing might help in this regard. Adding the ability to stop or speed up any animation in progress may also be useful.

Other additions to Viewpoints may include a help mechanism and supplementary materials. These could include suggestions for curriculum-related applications and goal-oriented lessons for students to direct to them in their explorations.

Furthermore, a thorough investigation of Viewpoints' effectiveness in teaching spatial skills could be pursued, as could an exploration into whether it may be beneficial for autistic children. It may inspire other instructional programs that deal with spatial ability. Most importantly, it may encourage more teaching of spatial abilities in school.

REFERENCES

Abrams, A. (1995). *Multimedia magic.* Boston: Allyn and Bacon.

Amorim, M. (2003). "What is my avatar seeing?": The coordination of "out-of-body" and "embodied" perspectives for scene recognition across views. *Visual Cognition, 10,* 157-199. Retrieved September 6, 2004, from http://ejournals.ebsco.com/direct.asp?JournalID=101901

Baxter, J. H., & Preece, P. F. W. (1999). Interactive multimedia and concrete three-dimensional modeling. *Journal of Computer Assisted Learning, 15,* 323-331. Retrieved September 6, 2004, from http://www.blackwellpublishing.com/journals/jca

Cohen, R. (1988). Formative evaluation of pre-Logo programming environments: a collaborative effort of researchers, teachers and children. *Journal of Computer-Based Instruction, 15,* 112-122.

Cohen, R., & Geva, E. (1989). Designing Logo-like environments for young children: the interaction between theory and practice. *Journal of Educational Computing Research, 5,* 349-377.

De Lisi, R., & Wolford, J. L. (2002). Improving children's mental rotation accuracy with computer game playing. *The Journal of Genetic Psychology, 163,* 272-282. Retrieved September 5, 2004, from http://pqasb.pqarchiver.com/heldref/

Dixon, J. P. (1983). *The Spatial Child.* Springfield, IL: C. C. Thomas.

Gerson, H. B. P., Sorby, S. A., Wysocki, A., & Baartmans, B. J. (2001). The development and assessment of multimedia software for improving 3-D spatial visualization skills. *Computer Applications in Engineering Education, 9,* 105-113.

Retrieved September 21, 2004, from http://www3.interscience.wiley.com/cgi-bin/jhome/38664

Hartley, T. (1998). *GUI design fundamentals: 1 day*. Phoenix, AZ: Computer PREP. Retrieved September 10, 2004, from http://www.netlibrary.com/

Mayer, R. E., & Fay, A. L. (1987). A chain of cognitive changes with learning to program in Logo. *Journal of Educational Psychology, 79,* 269-279.

McCuistion, P. J. (1991). Static vs. dynamic visuals in computer-assisted instruction. *Engineering Design Graphics Journal, 55,* 25-33.

Metros, S., & Hedberg, J. (2002). More than just a pretty (inter)face: The role of the graphical user interface in engaging eLearners. *The Quarterly Review of Distance Education, 3l,* 191-205.

Miller, C. L. (1992). Enhancing visual literacy of engineering students through the use of real and computer generated models. *Engineering Design Graphics Journal, 56,* 27-38.

Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology, 13,* 348-357.

Philleo, T. J. (1997). The effect of three-dimensional computer microworlds on the spatial ability of preadolescents (Doctoral dissertation, The Ohio State University, 1997). *Dissertation Abstracts International, 58,* 2611.

Piaget, J. P., & Inhelder, B. (1956). *The child's conception of space*. New York: The Humanities Press.

Rieser, J. J., Garing, A. E., & Young, M. F. (1994). Imagery, action, and young children's spatial orientation: it's not being there that counts, it's what one has in mind. *Child Development, 65,* 1262-1278.

Sachter, J. E. (1991). Different styles of exploration and construction of 3-D spatial knowledge in a 3-D computer graphics microworld. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 335-364). Norwood, NJ: Ablex Publishing.

Shiratuddin, N., & Landoni, M. (2002). Evaluation of content activities in children's educational software. *Evaluation and Program Planning, 25,* 175-182. Retrieved September 25, 2004, from http://www.sciencedirect.com/science/journal/01497189

Subrahmanyam, K., & Greenfield, P. M. (1994). Effect of video game practice on spatial skills in girls and boys. *Journal of Applied Developmental Psychology, 15,* 13-32. Retrieved September 25, 2004, from http://www.sciencedirect.com/science/journal/01933973

Tartre, L. A. (1990). Spatial orientation skill and mathematical problem solving. *Journal for Research in Mathematics Education, 21,* 216-229.

Tergan, S. (1998). Checklists for the evaluation of educational software: Critical review and prospects. *Innovations in Education and Training International, 35,* 9-20. Retrieved September 25, 2004, from http://ejournals.ebsco.com/direct.asp?JournalID=104709

Watson, J. A., Lange, G., & Brinkley, V. M. (1992). Logo mastery and spatial problem-solving by young children: Effects of Logo language training, route-strategy training, and learning styles on immediate learning and transfer. *Journal of Educational*

Computing Research, 8, 521-540.

Zavotka, S. L. (1987). Three-dimensional computer animated graphics: A tool for spatial skill instruction. *Educational Communication and Technology Journal, 35,* 133-144.

APPENDIX A

SUMMARY OF EDUCATORS' COMMENTS FROM ESE SYSTEM

Summary of Comments from ESE System

A) CONTENT
Teacher B: Most shapes are accurately named. I realize real sphere is not accurate due to program. More shapes possibilities.
Teacher E: The content is substantial.

B) INSTRUCTIONAL DESIGN
Teacher E: The instructional design appears thorough.

C) TECHNICAL QUALITY (Program Operation)
Teacher E: This is very complete. It was systematic and my student seemed to move through the system efficiently.

D) STUDENT USE (User Friendliness)
Teacher E: My student appeared to go through the program effortlessly. I heard him just once ask aloud if it was possible to change one of the images.

E) INSTRUCTOR USE
(no comments)

Best Aspects of the Software:
Teacher A: Unique viewpoint. Great experience.
Teacher B: Different viewpoints of 3-D software. Manipulation of figures.
Teacher C: It allows students to actually see things that lie in a plane.
Teacher E: Consistent but varied enough to maintain interest.

Worst Aspects of the Software:
Teacher A: No click and drag component.
Teacher B: Speed of program.
Teacher C: It can be difficult to manipulate.
Teacher E: Possibly a little too much imagery at one time at some points.

Potential Uses for the Software:
Teacher A: Solar system axis and rotation; planetary review and revolution.
Teacher B: Math - spatial figures. Basic animation.
Teacher C: Can be used in a math or science classroom. Can be useful in seeing thing from a new viewpoint.
Teacher E: From my point of view language, geometry, and fine arts.

Summary of Comments from ESE System (continued)

Estimate the amount of time students would need to use the program in order to achieve the objectives:
Teacher A: 45 minutes
Teacher B: 20-30 minutes
Teacher C: 1-2 hours
Teacher E: Because of its efficiency, students could achieve the objectives fairly quickly in most instances.

Final Recommendation Comments:
Teacher C: Can be useful used in a supplementary method in a classroom.
Teacher E A large number of students could benefit from this software. Many applications are evident.

Would you recommend it for purchase or not? Under what conditions?
Teacher A: Yes for spatial need!
Teacher B: With academic ideas for class - school purchase. Home use - neat program but students may become frustrated with speed.
Teacher C: Yes - as a supplementary program for classroom use.

Additional Question: Do you think this program helps students understand changes in point of view better?
Teacher A: Yes!
Teacher B: Yes.
Teacher C: Yes
Teacher D: Yes, and that is a huge consideration for an autistic child.

APPENDIX B

VIEWPOINTS ACTIVITY BOOKLET

Viewpoints

Activities Booklet for Viewpoints Application

Kathy Bateman

Texas A&M University
Masters Degree Program in Visualization Sciences

Welcome to Viewpoints!

Getting started – an empty World

When you first bring up the VIEWPOINTS application, you see a picture frame that has three arrows and a checkerboard "floor".

The green arrow represents up/down (the vertical axis).

The red and blue arrows represent left/right and forward/backward (the two horizontal axes).

This is an empty "World".

You can always start a fresh, new "World" like this one. To do this:
1. Choose the WORLD drop-down menu in the upper left corner.
2. Select "New" from this menu.

Activities in this booklet

You'll learn some new things with each activity you do. It's important to do them in numerical order.

These are the activities in this booklet:

1: SPINNING TOPS

Let's get started! Let's make spinning tops using VIEWPOINTS.
Here's what we'll do:

A. Make a box, move it, and change its color
B. Make a pyramid, move it, and change its color.
C. Spin the box.
D. Spin the pyramid.
E. Flip the pyramid over onto its point.
F. Spin the upside-down pyramid.

A. Make a box, move it, and change its color

First we'll create a box to spin so …

Make a box:
Find the "Object" pull-down menu at the top left side of the VIEWPOINTS window. Click and hold the mouse button and go down to "New", then choose the BOX object and click OK.

The box will appear in the middle of the screen. It will be in the middle of the Green, Red, and Blue axis arrows.

The "active object": Notice that the box has a thin yellow wire frame around it. This frame shows you which object is selected. This means that it is the object that will be affected by any of the OBJECT adjusters.

Move the box:
Now we'll move it up, so go to the OBJECT MOVE adjuster and click on the light green arrow two times.

Choose a color for the box:
Now change the color of the box by clicking on the OBJECT COLOR rectangle at the bottom right corner of window.

Pick a color, any color, and click OK.

B. Make a Pyramid, move it, and change its color

Now we'll make a pyramid using the same process as for the box.

Make a pyramid:
Find the "Object" pull down menu at the top left side of the VIEWPOINTS window and mouse down to "New", then choose the PYRAMID object and click OK.

The pyramid will appear in the middle of the screen and in the middle of Green, Red, and Blue axis arrows.

Move it:
Now we'll move it up, so go to the OBJECT MOVE adjuster and click on the light green arrow one time.

Change its color:
Now change the color of the pyramid by clicking on the OBJECT COLOR rectangle at the bottom right corner of window. Pick a color, any color, and click OK.

C. Spin the Box

Now the box should be balanced on top of the pyramid.

To make the box spin on top of the pyramid, make the box the active. To do this: Choose the object by clicking on it. The yellow wire frame should appear around the box.

Go to the OBJECT ROTATE adjuster and set the degrees to 90.

Now click either the light or dark green cone in the OBJECT ROTATE adjuster 5 or 6 times and watch it spin.

D. Spin the Pyramid

Now let's make the pyramid spin upside down on top of the box.

Make the box the active object if it is not already. Move it down one unit by clicking once on the dark green arrow in the OBJECT MOVE adjuster.

Can't click on it?
Hint: Try using the selector in the Object drop-down box or the choose object drop down box below the words "Current Object"

Move the pyramid up one unit by making it the active object and clicking once on the light green arrow in the OBJECT MOVE adjuster.

Now the pyramid is sitting on top of the box but it is on its base and it needs to be balanced on its point

E. Flip the Pyramid over onto its Point

Click on the pyramid. Make sure the OBJECT ROTATE adjuster is set to 90.degrees.

Now click the light blue cone 2 times and the pyramid will flip over and balance on its point (180 degrees).

F. Spin the Upside-Down Pyramid

Now click the either the light or dark green cone in the OBJECT ROTATE adjuster 5 or six times and watch it spin.

2: BALLS & RINGS

Let's start this activity with a fresh, empty world. To do this, go to the WORLD pull down box in the far upper left corner of the VIEWPOINTS window and select NEW.

In the rest of the activities of this lesson we'll type NEW WORLD when we want you to start over with an empty world.

Create a new sphere, color it, and move it

Now create a new sphere in the same way you made a box and a pyramid in the previous activity. Color it something pretty and move it up 1 unit just like you did in the previous activity. In future lessons, we'll use some "shorthand notation" for this task like:

NEW OBJECT - SPHERE
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
CONFIRM UNITS set to 1
CLICK LIGHT GREEN ARROW 1 time

OK, here is some more "shorthand notation" for the next task:

NEW OBJECT - RING
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
CONFIRM UNITS set to 1
CLICK LIGHT BLUE ARROW 2 times
CLICK LIGHT GREEN ARROW 1 time

Make the ring pass through the sphere

Ok, now its time for the fun to begin! You can get the ring to pass through the sphere by following the next step:

CHANGE OBJECT MOVE
CONFIRM UNITS set to 5
CLICK DARK BLUE ARROW 1 time

Make the ring larger

Now we make the ring larger so the sphere will pass through the hole in the middle of the ring.

CHANGE OBJECT SIZE
CONFIRM UNITS set to 1
CLICK LIGHT GREEN TRIANGLE 1 time
CLICK LIGHT RED TRIANGLE 1 time

Make the ring swallow the sphere

Now watch the ring swallow the sphere:

CHANGE OBJECT MOVE
CONFIRM UNITS set to 5
CLICK LIGHT BLUE ARROW 1 time

Put the ring over the sphere

Put the ring over the sphere in this next step:

CHANGE OBJECT MOVE
CONFIRM UNITS set to 1
CLICK DARK BLUE ARROW 2 times

Spin the ring around the sphere

Now spin the ring around the sphere:

CHANGE OBJECT ROTATE
CONFIRM UNITS set to 90 degrees
CLICK LIGHT BLUE CONE several times in succession

Make the sphere spin inside the ring

Now make the sphere spin inside the ring. Activate the sphere by clicking on it and then click some combination of Object Rotate Cones in rapid succession.

Now activate the ring again and do some other crazy combination of clicks on the Object Rotate Cones.

3: TELESCOPE

Let's make something new!

NEW WORLD
NEW OBJECT - RING
CHANGE OBJECT COLOR
CHANGE OBJECT SIZE
CONFIRM UNITS set to 10
CLICK LIGHT BLUE TRIANGLE 1 time

Bingo! You've just made yourself a telescope!

Looking through your telescope

To look through your telescope you'll need to swing it around a little bit. When it's perfectly lined up, you'll see the green axis arrow in the middle and all edges around your eyepiece will be equal.

Here's how to swing the telescope around.

To make "coarse" adjustments on the telescope's position, set the units to 10 on the OBJECT ROTATE adjuster, then click on the light green cone 3 times.

To make "fine" adjustments change the OBJECT ROTATE units to 1 and click the light and dark green cones until you get the best alignment.

When you can see through the telescope most directly, look at the OBJECT ROTATE color numbers. Red and blue should each be 0 degrees and green should be 32 degrees.

Moving the green arrow out of your way

Don't want to see that green arrow in the middle of your telescope view? Move your telescope up the green axis by clicking once on the OBJECT MOVE GREEN arrow (light green).

4: DOCKING

Set up this new situation:

NEW WORLD
NEW OBJECT - RING
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
  CONFIRM UNITS is set to 1
  CLICK LIGHT GREEN ARROW 2 times
  CLICK LIGHT BLUE ARROW 2 times

NEW OBJECT - CYLINDER
CHANGE OBJECT COLOR
CHANGE OBJECT ROTATE
  CONFIRM UNITS set to 90 degrees
  CLICK LIGHT RED CONE 1 time
CHANGE OBJECT MOVE
  CONFIRM UNITS set to 1
  CLICK LIGHT GREEN ARROW 2 times
  CLICK DARK BLUE ARROW 1 time

RING
CHANGE OBJECT SIZE
  CONFIRM UNITS set to 1
  CLICK COLORED TRIANGLES until you see 4 Blue, 2 Green, and 2 Red

<u>Docking the cylinder</u>

Now it is time to dock the solid cylinder inside the even larger hollow cylinder.

RING
CHANGE OBJECT MOVE by clicking the dark blue axis arrow 3 times.

(Make some deep throated scraping metallic sounds as you watch.)

5: GO STAND ON YOUR HEAD!

In this activity you will learn how to move around in the VIEWPOINTS world.

Whenever you start a new World you start out with the same default view of the World. This view is called the Mr. Default. Mr. Default is blue because he's an old stick-in-the-mud who never moves. But that's OK because it may be helpful to you that he is always in the same place. After you get your own person view, you may get lost and disoriented. If you do, you might find it helpful to go back and look at things from Mr. Default's point of view.

CLICK on any of the adjusters on the PERSON side of the window. Notice that none of them work. Mr. Default isn't going anywhere.

Becoming a new person

If you become your own person, you can move around and look around, too. To be a new person, click on the "Person" drop-down box at the top left of the window and select "New". Now the "Current Person" will be "Person 1". That's you!

You need to personalize and accessorize yourself. Use the "Person" drop-down box again and this time select "Rename". Click on "Person 1" in the "Rename as" and type "Me" then click "OK".

Now choose a personal color for yourself in the "Person Color" box at the bottom left of the VIEWPOINTS window.

Now you are dressed and ready to step out into the world as Me, oops! I mean, you, no, I mean you as Me, but not you as me because then I'd have to be you. Oh heck! You know what I mean, don't you?

Far out!

Go to the "Teleport" adjuster and change the units to 5, then click 3 times on the light yellow colored arrow.

What happened? BAM! You smashed right through that green arrow! And now you're on the other side of the arrow looking out into empty space.

Teleporting is similar to the "Person Move" adjuster except with "Person Move" you can only move in one plane at a time (blue, green, or red). Teleport allows you to move in all three dimensions at once. Just look in the direction you want to go and you can move toward or away from the location. This is very powerful for getting to a new location quickly.

Looking around

There are two ways to look around:

- You can use the "Look Around" adjuster to move your view (eyes) up, down, left or right or even at an angle.
- You can use the "Look At" feature in the "Person" drop-down box and select another person or object to look at.

Let's look at Mr. Default. Click the "Person" drop-down box in the upper left corner of the VIEWPOINTS window and select "Look at", then click on "Mr. Default" and click "OK".

Be patient - notice that after Me thinks about this for a moment, Me turns around and looks back at Mr. Default. Mr. Default is partially hidden behind the green arrow that Me smashed through (the arrow is mostly dark because it is facing away from the light source).

Me needs to move over just a little bit to get a clear view of Mr. Default. Make sure the PERSON MOVE units setting is 1 and then click 1 time on the light red arrow. Me moved over but Me didn't turn his body so Mr. Default has moved closer to the edge of the view. Turn Me's body a little bit using the "Look Around" adjuster so Mr. Default is back in the center of the view (click about 5 times on the dark purple arrow).

What does Mr. Default see?

So does Mr. Default see Me?

To find out what Mr. Default sees, you can choose either of these ways:

- Click the "Person" drop-down box and select "Go to…" and select Mr. Default.
- Click the "Current Person" drop-down box and click on Mr. Default.

You, but not Me, will "fly" back to Mr. Default and see through Mr. Default's "eyes".

From Mr. Default's view, Me is near the green arrow.

Make Me the current person again and you'll "fly" back to Me's point of view.

Cloning Me

Did you know you can clone Me? All you do is select "Person" drop-down box, select "Copy ..." in the same drop-down box. Clone yourself now. Change the color of your copy of Me and then rename "Copy of Me" to "Me Too!". (Sorry, we can't name him "Mini-Me" because he's the same size as Me.)

Me and Me Too! are good buddies but even good buddies need a little space, so do the following:

CHANGE PERSON MOVE
CONFIRM UNITS set to 1
CLICK DARK RED ARROW 2 times

Checking the view again

Now is a good time to see how things look from Mr. Default's viewpoint but before you go look, try to visualize in your head what Me and Me Too! will look like to Mr. Default.

Will Me Too! be on the right or left of Me from Mr. Default's view?

Time for Gymnastics!

Mr. Default loves to watch gymnastics because he can't do any himself. Let's make Me and Me Too! do a couple of tricks for Mr. Default.

Make Me the current person and do the following:

CHANGE LOOK AROUND
  CONFIRM UNITS set to 90 degrees
  CLICK LIGHT BLUE CURVED ARROW 2 times

Now what do you think Mr. Default sees? Take a guess and then go look from Mr. Default's view.

Wow, Me standing on his head is quite a trick.

Another trick for Mr. Default

Let's do another trick for Mr. Default. First make Me the current person and do this:

CURRENT PERSON - ME
CHANGE LOOK AROUND
  CONFIRM UNITS set to 90 degrees
  CLICK LIGHT BLUE CURVED ARROW 1 time

Now make Me Too! the current person and do this:

CURRENT PERSON – ME TOO!
CHANGE LOOK AROUND
  CONFIRM UNITS set to 90 degrees
  CLICK LIGHT BLUE CURVED ARROW 1 time

Got any ideas on what these two gymnasts are doing? Think for a minute and then look from Mr. Default's point of view.

Ta Da!

6: GOLF

Here's the setup:

NEW WORLD
NEW OBJECT – RING
CHANGE OBJECT COLOR
CHANGE OBJECT SIZE
    CLICK COLORED TRIANGLES to get 2 Red, 2 Green, and 1 Blue
CHANGE OBJECT ROTATE
    CONFIRM UNITS set to 90 degrees
    CLICK LIGHT RED CONE 1 time
CHANGE OBJECT MOVE
    CONFIRM UNITS set to 1
    CLICK LIGHT & DARK BLUE AND LIGHT & DARK RED.
        Stop clicking when you can't see the ring from the default view.

NEW OBJECT – SPHERE
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
    CONFIRM UNITS set to 1
    CLICK LIGHT GREEN ARROW 2 times.

NEW PERSON
RENAME PERSON to ME
CHANGE PERSON TELEPORT
    CONFIRM UNITS set to 10
    CLICK DARK YELLOW ARROW 1 time
CHANGE PERSON MOVE
    CONFIRM UNITS set to 1
    CLICK LIGHT RED ARROW 2 times
CHANGE PERSON LOOK AROUND
    CONFIRM UNITS set to 5
CLICK LIGHT PINK ARROW 1 time

<u>Going up!</u>

Now, let's go up an imaginary elevator:

CHANGE PERSON MOVE
    CONFIRM UNITS set to 5
    CLICK LIGHT GREEN ARROW 1 time

<u>Look at the ring from this viewpoint</u>

To see the ring from this viewpoint you'll have to look down.

CHANGE PERSON LOOK AROUND
    CONFIRM UNITS set to 5
    CLICK BROWN ARROW 3 times

Do you see the ring? Use the PERSON LOOK AROUND left and right arrows to move Me a little bit to either side.

<u>It's time for Golf!</u>

Now its time for a golf game.

Activate the sphere and use the light & dark red and light & dark blue object arrows to move the ball until you think it is directly over the ring.

When you think you have it lined up properly click the OBJECT MOVE dark green arrow 2 times to drop the ball into the ring.

Good Luck!

7: DEEP DIVING RING TOSS

Here's a new setup:

NEW WORLD
NEW OBJECT – PYRAMID
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
  CONFIRM UNITS set to 10
  CLICK DARK GREEN ARROW 1 time
  CONFIRM UNITS set to 1
  CLICK a random combination of LIGHT & DARK RED ARROWS and LIGHT & DARK BLUE ARROWS 8 times

NEW PERSON
RENAME PERSON to ME
CHANGE PERSON TELEPORT
  CONFIRM UNITS set to 5
  CLICK DARK YELLOW ARROW 1 time
CHANGE PERSON MOVE
  CONFIRM UNITS set to 1
  CLICK DARK GREEN ARROW 2 times

NEW OBJECT – RING
CHANGE OBJECT COLOR
CHANGE OBJECT ROTATE
  CONFIRM UNITS set to 90 degrees
  CLICK LIGHT RED CONE 1 time

Now you have an inner tube floating on the "surface" which you are looking at from below the surface.

The challenge

OK, here is a real challenge. It is similar to the challenge faced by deep-sea divers, which is "Just exactly where is that thing I'm looking at?" You don't have to contend with the distortion water creates for divers, but you will have to contend with the disorientation for lack of a reference.

The challenge is to drop the ring directly over the top of the pyramid that is some distance below and away from your view. Are you ready?

Here's how to start:

- CLICK PERSON LOOK AROUND ARROWS (mostly the light brown arrow) and find the pyramid in the water below.
- Now make sure the RING is the active object.
- Using only the OBJECT MOVE adjuster and the PERSON LOOK AROUND adjuster try to drop the ring over the top of the pyramid.

Good Luck!

8: FLY TO THE TOP OF THE BOX

New setup:

NEW WORLD
NEW OBJECT - BOX
CHANGE OBJECT COLOR
CHANGE OBJECT SIZE
CLICK LIGHT GREEN TRIANGLE 6 times
CONFIRM 1 Red, 1 Blue, and 6 Green
CHANGE OBJECT MOVE
CLICK COLORED ARROWS until you have –5 Blue and 3 Green

NEW PERSON
RENAME PERSON to ME
CHANGE PERSON LOOK AROUND
CONFIRM UNITS set to 45 degrees
CLICK LIGHT BLUE ARROW 1 time
CHANGE PERSON TELEPORT
CONFIRM UNITS set to 10
CLICK DARK YELLOW ARROW 1 time

<u>The challenge</u>

Are you up to this?

From this viewpoint your challenge is to fly (move) directly over the top of the box and look "down" so all you can see from your view is the top of the box and the checkerboard floor below.

Sound easy? Here are a few restrictions on which adjusters you can use.

- You can use the PERSON TELEPORT adjuster
- You can use the PERSON MOVE adjuster
- You can use only the <u>straight arrows</u> on the PERSON LOOK AROUND adjuster.

Good Luck, Captain

9: LOST IN SPACE

What's up? What's down?

These are serious questions for astronauts floating in space without any reference point. This activity may give you an idea of the difficulties astronauts face when working in outer space.

Follow the setup instructions below to put everything in place before the game begins. Don't worry if you don't physically see these objects while you are using the adjusters – that's just part of the fun ahead.

NEW WORLD
1. Go to the "World" drop-down box
2. Select the "Ground Plane". (This will turn off the ground plane. In this program, "plane" is used in its geometry definition, not as a flying object.)

NEW OBJECT - RING
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
  CONFIRM UNITS is set to 5
  CLICK DARK GREEN ARROW 2 times
  CLICK DARK RED ARROW 2 times
  CLICK LIGHT BLUE ARROW 1 time
CHANGE OBJECT SIZE
  CONFIRM UNITS set to 1
  CLICK COLORED TRIANGLES until you see 4 Blue, 2 Green, and 2 Red

Check your values against this table.

Object Move		
	Red	-10
	Green	-10
	Blue	5
Object Rotate		
	Red	0
	Green	0
	Blue	0
Object Size		
	Red	2
	Green	2
	Blue	4

NEW OBJECT - CYLINDER
CHANGE OBJECT COLOR
CHANGE OBJECT MOVE
CONFIRM UNITS set to 1
CLICK LIGHT GREEN ARROW 3 times
CLICK DARK RED ARROW 7 times
CLICK DARK BLUE ARROW 5 times
CHANGE OBJECT SIZE
CONFIRM UNITS set to 1
CLICK COLORED TRIANGLES until you see 1 Blue, 6 Green, and 1 Red

Check your values against this table.

Object Move		
	Red	-7
	Green	3
	Blue	-5
Object Rotate		
	Red	0
	Green	0
	Blue	0
Object Size		
	Red	1
	Green	6
	Blue	1

NEW PERSON
RENAME PERSON to ME
CHANGE PERSON COLOR
CHANGE PERSON TELEPORT
CONFIRM UNITS set to 5
CLICK LIGHT YELLOW ARROW 3 times
CHANGE PERSON MOVE
CONFIRM UNITS set to 5
CLICK LIGHT GREEN ARROW 5 times (this will take a little while)

Check your values against this table.

Person Move		
	Red	-3
	Green	26
	Blue	-5
Person Rotate		
	Red	0
	Green	32
	Blue	0

The Challenge

As Astronaut Me, you are floating in space.

The Federation Intergalactic Command Center has radioed a very important assignment to you. A Hyperforce Induction Ring (HIR) has broken loose from an Interplanetary Transfusion Cylinder (ITC) and both the ring and the cylinder are drifting apart into space.

Your job is to push the ITC (cylinder) back to the HIR (ring) and slide the ITC through the HIR until it is again mounted directly over the center of ITC.

You control the ITC so you may use the Object Move and Object Rotate adjusters on the ITC but the HIR is free floating so you can't change anything about the HIR.

Astronaut Me can go and can look anywhere to make sure everything is lined up perfectly before sliding the cylinder through the hole in the center of the HIR.

Your assignment is done when the HIR is centered on the ITC.

Good Luck, Astronaut Me!

Here are some hints:

Hint #1: Use the "Look at" function in the "Person" drop-down box to look at one object and then look at the other object to determine where they are in relation to Astronaut Me.

Hint # 2: Keep Astronaut Me near the ITC as you move it towards the HIR.

Hint #3: To determine when the ITC is lined up and ready to slide through the HIR, move Astronaut Me to the far end of the ITC and look down the length of the ITC toward the HIR.

Hint #4: Is Astronaut Me really lost in space? If so, go to the "World" drop-down box and select "Ground Plane". Maybe this will help?

Thank you for participating in these Viewpoints activities!

VITA

Kathleen Suzanne Bateman

450 Roundup Trail

Austin, TX 78745

Bachelor of Arts in Studio Art

Grinnell College

May 1997

Master of Science in Visualization Sciences

Texas A&M University

December 2004

Animation Artist

Engineering Animation, Inc.

Summer 1998

Real-time Cinematic Tech

Digital Anvil

January 2001 – October 2002