

Efficient, Intuitive User Interfaces for Classroom-Based Immersive Virtual Environments

Doug A. Bowman, Matthew Gracey & John Lucas
Virginia Tech, Dept. of Computer Science, Blacksburg, Virginia, USA
{bowman | mgracey | jolucas}@vt.edu

Abstract

The educational benefits of immersive virtual environments (VEs) have long been touted, but very few immersive VEs have been used in a classroom setting. We have developed three educational VEs and deployed them in university courses. A key element in the success of these applications is a simple but powerful user interface (UI) design that requires no training, yet allows students to navigate through and interact with the virtual world in meaningful ways. We discuss the design of this UI and the results of an evaluation of its usability in university classrooms.

1. Introduction

For many years, researchers in the area of immersive virtual environments (VEs) have pointed to education as a key application area. In his seminal 1965 article “The Ultimate Display,” Ivan Sutherland stated that if a three-dimensional (3D) graphics display could be controlled by head and body movements (an immersive VE), then the display could be “a looking glass into a mathematical wonderland” [15]. In other words, Sutherland realized that the visualization of data in an immersive environment could enhance understanding and learning. In the National Research Council’s report on virtual reality, the use of VEs in education was also considered a crucial area for future research [7]. The report envisioned the use of VEs to teach high school students on a worldwide network.

VEs can allow students to visualize and interact with complex three-dimensional (3D) structures, perform virtual “experiments,” view scenes with natural head and body movements, and experience environments that would be otherwise inaccessible because of distance (the surface of the Moon), scale (a complex molecule), or danger (a sunken ship). Despite the predictions of VEs in the classroom, however, the use of immersive VEs in educational settings has been limited. High cost and low of usability have impeded the adoption of these systems.

In our research, we have been developing educational applications of VEs with a focus on usability and cost. We take advantage of emerging results from the field of three-dimensional user interfaces (3D UIs) [4], and

carefully design and evaluate the UI metaphors and interaction techniques for our applications. We also put together low-cost, portable immersive VE systems designed for classroom use. We take the approach of immersing one student at a time and allowing the rest of the class to see the VE through that student’s eyes.

Our prior experience in this area has led us to postulate four principles for designing educational VEs:

1. *Interactivity*: Educational VEs should provide students many opportunities to interact with and explore the environment, rather than simply offering a static visualization.
2. *Complementarity*: VEs should be used to complement, not replace, other methods of teaching and learning in the classroom.
3. *Information-Richness*: Educational VEs should contain many different types of information (geometric, textual, auditory, concrete, abstract, etc.).
4. *Augmentation*: Educational VEs will be most effective when they portray information that is not normally visible or accessible in the physical world.

In this paper, we focus on the first of these principles (*interactivity*). We believe that educational VEs need to be highly interactive in order to succeed, because interactivity can allow students to test hypotheses; it can allow for individualized experiences for each student; and it can increase engagement. If highly interactive VEs are to be used in the classroom, however, the UI must be transparent – instantly understandable and extremely efficient. Usability is not sufficient to assure student learning, but it is necessary.

We describe a generic UI metaphor that has proven to be highly usable in real classroom use with three different educational VE applications. We begin by discussing related work and the three target applications. Next, we describe the hardware and software components of our UI design, and present usability results from student use.

2. Related work

Wickens [17] argues that the goals of user interface design (e.g. reduce mental workload for the user) and educational software design (e.g. force the user to work in

order to learn) conflict. We argue that a distinction needs to be made between cognitive load from task-related activities and that from system-related activities. In educational environments, learning activities (task-related) should require effort and choice; however, system-related activities, such as finding a menu item, should require little cognitive processing. One does not want the interface to be a distraction from learning.

There have been some moderately successful uses of VEs in classroom settings, such as ScienceSpace [14], the Virtual Habitat [6], and several applications by Johnson and his colleagues (e.g. [9]). Effective UI design, however, was not an explicit focus of most of this prior work.

Much of our work makes use of recent research in 3D interaction and 3D UIs [4]. 3D UI researchers have taken principles from the human-computer interaction (HCI) literature and applied them to VEs and other 3D environments. Such research includes the design of interaction techniques for tasks such as travel (e.g. [16]), object selection and manipulation (e.g. [13]), and system control (e.g. [1]). We also draw from the literature on the evaluation of VE interfaces [2], and we subscribe to a usability engineering approach for 3D UIs [8].

3. Educational VE applications

In this section we briefly describe the three immersive VE tools we have developed for classroom use. Each of the applications has been used and evaluated in actual classrooms, with students using the VE systems under the guidance of the instructor. Only the functionality of the applications is presented here; the UI metaphor and interaction techniques are discussed in the next section.

3.1 Virtual-SAP

Virtual-SAP [5] is a VE tool for visualizing the effects of earthquakes on building structures. Users of Virtual-SAP can build 3D structures (figure 1a) from columns, beams, slabs, etc., and can specify the size and material properties of each element. The response of the structure to an earthquake is then simulated with the Structural Analysis Program (SAP), and the results are shown as a 3D animation within the VE.

Architecture and engineering students can use Virtual-SAP in at least two ways to enhance their understanding. First, they can watch from any point of view as pre-defined structures are subjected to earthquake loads. Second, they can test various “what-if” scenarios, either by modifying an existing structure or building one from scratch. This allows students to construct their own mental models of the interaction between the structure and the earthquake.

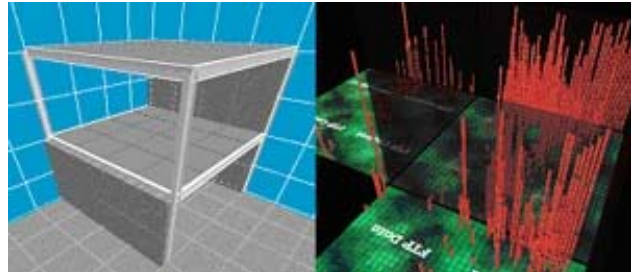


Figure 1. a) Building Structure in Virtual-SAP. b) Overhead view in NetViz

3.2. VENTS

The Virtual Environment Normalizing Transformation System (VENTS) helps computer graphics students learn the normalizing transformation for 3D perspective views. The normalizing transformation is a multi-step process that takes an arbitrary 3D viewing situation (camera position and orientation, view window size, clipping plane information) and transforms it into a canonical viewing situation, in order to simplify the process of perspective projection.

VENTS allows students to visualize a 3D world coordinate system, a view frustum along with related clipping planes, points, and vectors, and an object to be rendered (figure 2). Students can then select the various steps of the transformation, and see the objects move to show the effect of each step. Students can navigate to any location in order to fully understand the 3D nature of the transformations. Audio descriptions of each step are available, and users can interactively set the initial conditions for the transformation.

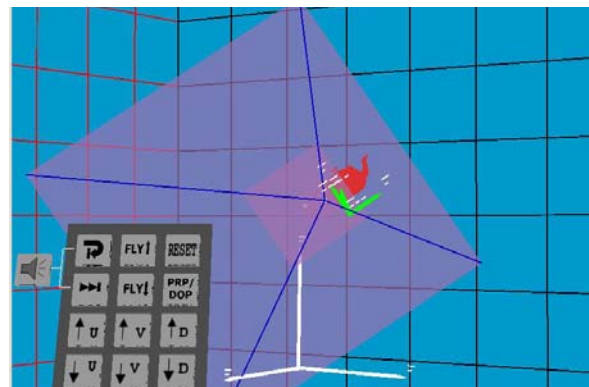


Figure 2. User's initial view in VENTS

3.3 NetViz

The third educational VE, NetViz, is a visualization of large-scale network traffic data. This tool is intended to help students visualize an abstract, multi-parameter dataset (traces or real-time data from network traffic). NetViz (figure 1b) uses a “city” metaphor to represent the

data, with buildings representing a connection or set of connections. Students can watch the landscape change as time passes, navigate to any point within the space, and analyze patterns that emerge from the data.

Instructors have used NetViz to illustrate various concepts by guiding students to particular locations in the VE, and then asking the class to interpret the information being displayed. According to these instructors, it would be nearly impossible to explain some of these concepts without the help of the visualization tool.

4. User Interface Design

We have designed a standard UI for these three applications focused on the requirements of classroom-based VEs. Although each of the applications has some unique interface components, the overall UI metaphor, interaction style, and input/output devices are the same across all three applications. In this section, we present our UI design goals, the hardware we use in our systems, and the various components of the UI.

4.1. Goals

All immersive VEs need usable interfaces that provide users with good affordances and feedback, prevent user fatigue, and keep users from becoming disoriented. VEs deployed in classrooms, however, have additional usability requirements. We designed our UI with these goals in mind:

- **Learnability:** Classroom time is always very limited and very precious. Time spent learning the interface cannot be used for learning content, so VEs must be instantly usable by students (who may have never used a VE before).
- **Efficiency:** We want students to interact with the VE to enhance their learning experience. This interaction must be achieved, however, in the shortest possible amount of time. This requires that the interface must be transparent, allowing immediate access to the educational content.
- **Multiple student use:** We want as many students as possible to use the immersive VE directly, since we believe that immersion and direct control offer more opportunities for learning.
- **Whole class benefits:** In a medium- or large-sized class, it will be impossible to have all students use the system directly during class time, so our UI needs to allow non-immersed, passive students to also obtain most of the same learning benefits.
- **Ease of use for instructors:** Teachers want to use the VE as a teaching tool, rather than simply

watch students use the VE. Thus, teachers need to be able to direct the student using the VE, point things out to the class as a whole, and to control certain aspects of the system.

4.2. Hardware

The hardware we chose for our system needed to be inexpensive, so that schools could afford it, and portable, so that it could be taken to the classroom, rather than forcing classes to come to a research lab. Because of these requirements, we chose to put together the hardware components of our UI first (except for the choice of primary input device). This is in contrast to our preferred method of letting the UI requirements and high-level UI design drive the choice of hardware.

We assembled a system using only commercial, off-the-shelf (COTS) components, consisting of:

- Dual-boot (Windows 2000/Linux) PC with an NVIDIA GeForce2 graphics card.
- Daeyang i-visor head-mounted display (HMD) with 800x600 resolution, a 30-degree horizontal field of view (FOV)
- Intersense Intertrax2 3DOF tracker
- Handykey Twiddler2 12-button chord keyboard
- InFocus LCD projector
- Rolling A/V cart

These components assembled give us a portable immersive platform for use in the classroom. The system allows for one fully immersed user and displays the immersed user's view via the projector for a classroom of non-immersed students. This complete immersive VE system cost less than \$7000.

Of course, we made many tradeoffs in order to achieve such a low-cost portable system. First, the system does not have the capability to display stereo graphics through either the HMD or projector, so our applications run in biocular (same image to both eyes) mode in the HMD, and monoscopic mode on the screen. Second, only the user's head is tracked, so all spatial input must come from the head. Interaction techniques that rely on a tracked hand (or two) cannot be used. Third, the tracking is only 3DOF, meaning that users cannot translate their viewpoint by walking, stooping, leaning, etc. Lastly, the HMD suffered from a low FOV and various ergonomic issues.

4.3. UI design components

The hardware components described above place some constraints on the type of UI that can be designed. The user's only input devices are the 3DOF head orientation tracker and the chord keyboard. The keyboard is limited to 12 buttons. All visual feedback must be displayed on

the HMD, which has decent resolution, but a very limited FOV. Audio feedback is also possible.

The choice of input, output, and supporting devices is an important part of UI design, but is not the only crucial component. Within the constraints of the hardware, we still had plenty of room for creative UI and interaction design.

4.3.1. Remote control metaphor. Our UI design is based on a “remote control” metaphor. Like most common remote controls, we wanted to map each application function to a dedicated button on the input device. This leads to easy learnability and prevents errors [11]. In our design, the user holds the chord keyboard like a remote control and uses only the thumb to locate and press buttons, although this was not intended by the designers of the chord keyboard (figure 4).

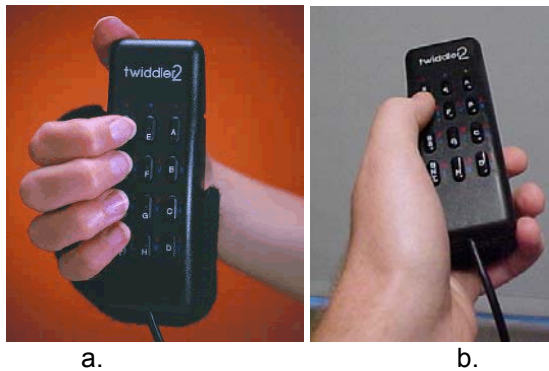


Figure 4. a) typical chord keyboard grip; b) grip used in the “remote control” metaphor

We present a visual aid (the *interface representation*) in the corner of the user’s view reminding them of the mapping between buttons and functions (figure 5 shows a



Figure 5. Close-up of interface representation in Virtual-SAP

close-up, see figure 2 for actual size of interface representation on the display).

4.3.2. Button overloading. In Virtual-SAP, there are too many possible actions to use a simple one-to-one mapping between buttons and functions. To allow access to all of these functions, we designed a virtual “tabbed” interface. The top three buttons are used to select a tab (mode), and the remaining nine are used for functions within the current tab (see figure 5). We also used context-sensitive mappings to lessen this problem (e.g. the “select” button changes to “drop” when an object is being manipulated).

4.3.3. Menu lists. In some cases, the user wishes to select an item from a list, such as the type of structural element she wants to create in Virtual-SAP. To achieve this, we provide a single button that cycles through the available items in the menu. The currently selected item appears in a text box next to the interface representation.

4.3.4. 3D navigation. VE navigation involves viewpoint orientation and positioning. Users turn their heads naturally to view the 3D world, and the view is calculated based on the head tracker data. To translate the viewpoint, we provide “fly forward” and “fly backward” buttons (see figure 5). These functions are always available, even when the user changes tabs in Virtual-SAP. The direction of travel is along the user’s head-orientation vector (gaze-directed steering) [10].

4.3.5. 3D selection and manipulation. We also use the tracker information for object selection and manipulation. In Virtual-SAP and NetViz, a reticle (crosshair) appears in the center of the screen. A ray is cast from the eye through this point, and the user may select the first object intersected by the ray. For objects that can be moved, like the structural elements in Virtual-SAP, the user turns her head while the object is selected to keep the object on the ray, and can also change the depth of the object along the ray using two buttons on the input device (the “Obj” buttons in figure 5). When an object is being manipulated, two buttons on the input device are used for rotation about its’ principal axis.

4.3.6. View management. We provide a “view locking” feature in our applications, which essentially disables the head tracker momentarily, allowing the user to move her head without changing the view. This can be important when an instructor wants to point out something in a particular view – keeping the view stable while eliminating the need for the user to hold her head in a fixed position. NetViz also contains a set of pre-defined user viewpoints, allowing the instructor to take the user

smoothly from their current location to another important viewing position.

5. Results

We have used our educational VE applications in five different undergraduate and graduate classes. The classes ranged in size from 7 to 45 students. We collected various measures of usability based on observations of student behavior, surveys, and post-experience interviews. Here we focus on usability, defined broadly to include user task performance, user preferences and satisfaction, user comfort, etc.

5.1 Overall usability

Overall, the usability of the applications was high. Students were given only minimal instructions at the beginning of each class, but all students who used the applications directly were able to navigate the 3D environment and perform tasks within it. In our most recent classroom visits, students who used Virtual-SAP rated it at 6.67 on a 7-point scale for ease of use. Users of NetViz had an average rating of 5.67.

The remote control metaphor was also successful. All students understood the mapping between the representation and the buttons on the device, and even when buttons changed based on the current mode, students had no difficulty adapting. It was sometimes difficult to find the correct physical button. This problem would likely be eliminated if we used a nine-button (3x3 layout) device, where the user could keep his thumb on the center button and then simply move in one of eight directions to find a button.

5.2 3D navigation

Gaze-directed flying was intuitive for most students, although many were surprised to find they could move up/down as well as horizontally. Typically, we would have to prompt students to look up in order to fly above the ground, for example. For sparse environments, the ability to fly backwards is critical, as it allows you to keep objects in view even when flying away from them.

5.3 3D selection and manipulation

We were concerned that head-based selection and manipulation would be difficult, since these tasks are normally done with the hand in immersive VEs. We observed, however, that students picked this up quite easily and used it effectively. Gaze-based selection is less fatiguing than image-plane selection [12, 3] using the hand, and gaze-based manipulation can be quite efficient

if a snap-to-grid technique is used (as in Virtual-SAP). One architecture student was able to build a complete two-story structure with walls and floors (approximately 80 separate elements) in about 5 minutes, making no selection or placement errors.

5.4 Audience comprehension

The combination of one immersed student and an “audience” of observing students was a potential problem area, but we found that the audience understood what they saw on the screen quite well. Using a 7-point scale, they rated their understanding of what they saw on the screen at averages of 5.75 (Virtual-SAP) and 5.42 (NetViz). We attribute this to the fact that we positioned the HMD user just to the side of the screen, so that other students could take in both the motions of the user and the user’s view in a single glance.

5.5 View management

The ability to lock the user’s view and to take the user to pre-defined viewpoints allowed instructors more control over what the class saw. In Virtual-SAP, for example, the instructor wanted students to view structures from above, so view locking allowed users to relax while seeing a bird’s-eye view. While the view was locked, the instructor could easily point out various features shown in the view.

5.6 Usability problems

The only major negative usability results relate to the low cost VE hardware we used. As noted above, our HMD was difficult to keep in place and uncomfortable to wear. The narrow FOV made it easy for users to become disoriented. The head tracker produced unpredictable data when the head orientation was out of its range (+/- 70 degrees pitch). This caused much confusion with the students and often required that the application be restarted.

Another area for further study is the communication between the instructor and the student using the VE. Since the student is using an HMD, the instructor cannot physically point out anything in the VE to the student. This leads to confusing conversations as the instructor attempts to guide the student verbally. A 3D cursor or virtual pointer for the instructor might help to solve this problem.

6. Conclusions and future work

We have presented a generic UI for immersive VE applications used in a classroom. This UI does not require

expensive hardware, and meets our stated goals (section 4.1):

- it can be learned almost instantly;
- it is efficient for navigation and interaction within a 3D environment;
- it allows multiple students to experience the VE during a class period;
- it allows the entire class to benefit; and
- it is generally easy for instructors to use as a teaching tool.

We found that the UI metaphor was extremely effective, but that low-cost devices can hinder usability to some degree. We can “design around” the limitations of these devices up to a point, but the constraints inherent in the hardware can still reduce the effectiveness of the applications.

We are continuing to develop and evaluate this UI. We plan to experiment with different display and input devices, additional navigation techniques, and other interface enhancements. Our main contribution is a UI design that does not get in the way of student learning – as discussed in section 2, we have provided an environment where cognitive resources can be allocated to learning tasks rather than interface tasks. Our ultimate goal is to provide educationally effective VEs, an objective that requires very high levels of usability.

7. Acknowledgements

This work was partially supported by National Science Foundation Grant DUE-0127326. We would like to thank Mehdi Setareh and Alex Kalita for their work on the Virtual-SAP project. We also thank Srinidhi Varadarajan and Mark Henry for help with the NetViz project. VENTS was originally developed and evaluated by Prasuna Chennupati and Kristin Wheeler.

8. References

- [1] I. Angus and H. Sowizral, "Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment," presented at *SPIE, Stereoscopic Displays and Virtual Reality Systems*, 1995.
- [2] D. Bowman, J. Gabbard, and D. Hix, "A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods," *Presence: Teleoperators and Virtual Environments*, vol. 11, 2002.
- [3] D. Bowman, D. Johnson, and L. Hodges, "Testbed Evaluation of VE Interaction Techniques," *Presence: Teleoperators and Virtual Environments*, vol. 10, pp. 75-95, 2001.
- [4] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev, "An Introduction to 3D User Interface Design," *Presence: Teleoperators and Virtual Environments*, vol. 10, pp. 96-108, 2001.
- [5] D. Bowman, M. Setareh, M. Pinho, N. Ali, A. Kalita, Y. Lee, J. Lucas, M. Gracey, M. Kothapalli, Q. Zhu, A. Datey, and P. Tumati, "Virtual-SAP: An Immersive Tool for Visualizing the Response of Building Structures to Environmental Conditions," presented at *IEEE Virtual Reality*, Los Angeles, 2003.
- [6] D. Bowman, J. Wineman, L. Hodges, and D. Allison, "The Educational Value of an Information-Rich Virtual Environment," *Presence: Teleoperators and Virtual Environments*, vol. 8, pp. 317-331, 1999.
- [7] Durlach, N. and A. Mavor, *Virtual Reality: Scientific and Technological Challenges*, National Academy Press, 1995.
- [8] J. Gabbard, D. Hix, and J. Swan, "User-Centered Design and Evaluation of Virtual Environments," *IEEE Computer Graphics & Applications*, vol. 19, pp. 51-59, 1999.
- [9] A. Johnson, T. Moher, S. Ohlsson, and J. Leigh, "Exploring Multiple Representations in Elementary School Science Education," presented at *IEEE Virtual Reality*, Yokohama, Japan, 2001.
- [10] M. Mine, "Virtual Environment Interaction Techniques," UNC Chapel Hill CS Dept., Technical Report TR95-018, 1995.
- [11] Norman, D., *The Design of Everyday Things*. New York: Doubleday, 1990.
- [12] J. Pierce, A. Forsberg, M. Conway, S. Hong, R. Zeleznik, and M. Mine, "Image Plane Interaction Techniques in 3D Immersive Environments," presented at the *ACM Symposium on Interactive 3D Graphics*, 1997.
- [13] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, "The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR," presented at the *ACM Symposium on User Interface Software and Technology*, 1996.
- [14] M. Salzman, C. Dede, R. Loftin, and J. Chen, "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning," *Presence: Teleoperators and Virtual Environments*, vol. 8, pp. 293-216, 1999.
- [15] I. Sutherland, "The Ultimate Display," presented at the *IFIP Congress*, 1965.
- [16] D. Tan, G. Robertson, and M. Czerwinski, "Exploring 3D Navigation: Combining Speed-Coupled Flying with Orbiting," presented at *ACM CHI*, Seattle, 2001.
- [17] C. Wickens, "Virtual Reality and Education," presented at the *IEEE International Conference on Systems, Man, and Cybernetics*, 1992.