Virtual Reality Interfaces Using Tweek

Patrick Hartling,* Allen Bierbaum,† Carolina Cruz-Neira‡
Virtual Reality Applications Center (VRAC), Iowa State University

1 Introduction

Developers of virtual environments often face a difficult problem: users must have some way to interact with the virtual world. The application developers must determine how to map available inputs (buttons, gestures, etc.) to actions within the virtual environment (VE). As a result, user interfaces may be limited by the input hardware available with a given virtual reality (VR) system.

To address such limitations, we have developed Tweek, a middleware tool that presents users with an extensible Java graphical user interface (GUI) capable of communicating with VR applications. Tweek can run on desktop computers, on palmtop computers in projection-based virtual reality systems, or in a three-dimensional virtual space. We use Tweek to address some limitations of input and interaction within virtual environments (VEs) by designing GUIs that utilize familiar two-dimensional (2D) components. Interaction techniques that require fine-grained control or high-precision input are often better suited to 2D interfaces than the use of gestures or buttons on a wand [Hill 2000].

The use of Java-based GUIs for interaction within VEs has been investigated in previous works at the VRAC. The first such work investigated the usability of a palmtop system with a Java GUI to interact with and manipulate a virtual space [Hill 2000]. Tweek extends this research by generalizing the interaction capabilities and the dynamic GUI component loading.

Tweek is a collection of multiple technologies: C++, Java, JavaBeans, and CORBA. Combined, these allow a Java GUI composed of plug-ins to communicate with a C++ application. Our implementation aims to simplify the inter-language communication so that programmers can make use of Tweek in their VR applications without knowing all the details of the individual technologies.

At the heart of the inter-language communication is CORBA, the Common Object Request Broker Architecture [OMG 2001]. It provides a cross-platform, language-independent method for distributed objects to communicate. In Tweek, it manages all communication between the Java GUI and the C++ VR application. Because CORBA is language-independent, there exists the potential for use of other languages besides Java and C++.

The Java GUI itself is a generalized framework that loads JavaBeans [JavaSoft 1997] dynamically using XML-based descriptions. The JavaBeans may encapsulate any functionality, but those that extend the GUI are crucial to the use of Tweek in a VE. Such graphical JavaBeans are written by the VR application developers and are customized for use with a given application. Dynamic extension of the GUI allows the VR application to add components to the interface while the user is active within the virtual space. Through dynamically extensible input, we can explore new possibilities for interactivity in VEs.

2 Implementation

Tweek utilizes an extensible object-oriented design based upon the Observer design pattern [Gamma et al. 1995]. The subject is the state of the VR application, and the observer is the GUI. The state information is specific to each application.

An example use of Tweek on the desktop is shown in Figure 1a. Here, a top-down navigation tool is available for point-and-click navigation within a three-dimensional (3D) space. The observed state is the current user position in the VE. Bookmarks can be selected for direct navigation to known locations, and the 2D map can be clicked to navigate to the selected location. The same navigation tool can be used in an immersive environment using a palmtop computer, shown in Figure 1b.

3 Conclusion and Future Work

At VRAC, we use Tweek on palmtop computers to control VR applications in our projection-based VR systems. To map Tweek into a 3D environment, we plan to experiment with existing work such as 3Dwm [Elmqvist 2001]. We hope to offer an immersive interface with familiar controls without re-inventing GUI technology.

References

ELMQVIST, N. 2001. *3Dwm: Three-Dimensional User Interfaces Using Fast Constructive Solid Geometry*. Master's thesis, Chalmers University of Technology, Göteborg.

GAMMA, E., HELM, R., JOHNSON, R., AND VLISSIDES, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison-Wesley Publishing Company, New York, NY.

HILL, L. C. 2000. *Usability of 2D Palmtop Interaction Device in Immersive Virtual Environments*. Master's thesis, Iowa State University, Ames, IA.

JAVASOFT. 1997. *JavaBeans 1.01 Specification*. Sun Microsystems, Mountain View, CA.

OMG. 2001. *The Common Object Request Broker: Architecture and Specification*, 2.6 ed. Object Management Group.

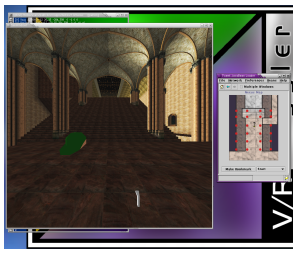

Figure 1: Tweek navigation on desktop (a) and PDA (b)

*email: patrick@vrac.iastate.edu
†email: allenb@vrac.iastate.edu
‡email: cruz@iastate.edu