

## Design Strategies and Functionality of the Visual Interface for Virtual Interaction Development (VIVID) Tool

Lac Nguyen  
I-Net Corporation

Patrick J. Kenney  
LinCom Corporation

Software Technology Branch  
NASA / Johnson Space Center

### ABSTRACT

Development of interactive virtual environments (VE) has typically consisted of three primary activities: model (object) development, model relationship tree development, and environment behavior definition and coding. The model and relationship tree development activities are accomplished with a variety of well-established graphic library (GL) based programs - most utilizing graphical user interfaces (GUI) with point-and-click interactions. Because of this GUI format, little programming expertise on the part of the developer is necessary to create the 3D graphical models or to establish interrelationships between the models. However, the third VE development activity, environment behavior definition and coding, has generally required the greatest amount of time and programmer expertise. Behaviors, characteristics, and interactions between objects and the user within a VE must be defined via command line C coding prior to rendering the environment scenes. In an effort to simplify this environment behavior definition phase for non-programmers, and to provide easy access to model and tree tools, a graphical interface and development tool has been created. The principal thrust of this research is to effect rapid development and prototyping of virtual environments. This presentation will discuss the "Visual Interface for Virtual Interaction Development" (VIVID) tool; an X-Windows based system employing drop-down menus for user selection of program access, models and trees, behavior editing, and code generation. Examples of these selections will be highlighted in this presentation, as will the currently available program interfaces. The functionality of this tool allows non-programming users access to all facets of VE development while providing experienced programmers with a collection of pre-coded behaviors. In conjunction with its existing interfaces and predefined suite of behaviors, future development plans for VIVID will be described. These include incorporation of dual user virtual environment enhancements, tool expansion, and additional behaviors.

### INTRODUCTION

The creation of interactive virtual environments (VE) generally requires three developmental steps: model creation, model relationship definition, and environment/object behavior definition. Of these three, the behavior definition necessitates the greatest amount of time and expertise for coding on the part of a programmer. To alleviate much of this dependence on a dedicated programmer, and to create a rapid prototyping tool for VE demonstrations, the Visual Interface for Virtual Interaction Development (VIVID) was developed at the Software Technology Branch (STB) of the NASA/Johnson Space Center. VIVID not only provides a common interface between existing graphical library development programs being used, but also generates the code associated with behaviors and actions in a virtual environment. This paper discusses the design and interface considerations addressed while developing VIVID, describes examples of its functional execution, and outlines existing capabilities and directions for future enhancements.

## BACKGROUND

Development of interactive virtual environments (VE) has typically consisted of three primary activities: model (object) development, model relationship tree development, and environment behavior definition and coding. The model and relationship tree development activities are accomplished with a variety of well-established graphic library (GL) based programs - most utilizing graphical user interfaces (GUI) with point-and-click interactions.

The GL development programs currently in use for VE development at the STB Lab are the Solid Surface Modeler<sup>1</sup> (SSM) and the Tree Display Manager<sup>1</sup> (TDM). SSM is a graphics application for solid shaded and wireframe 3D geometric modeling and is used to develop the models (objects) that comprise a VE. TDM is a graphics visualization tool which uses a hierarchical representation (relationship tree) of the 3D models created with SSM to give structure to a VE. It provides an interface for generating and manipulating the relationship tree of models needed to portray a visual scene. For example, in TDM the user will specify which models to include in the environment, the position and intensity of light sources, and the visible perspectives of the environment (e.g. eyes and cameras). Both the SSM and TDM programs were developed on Silicon Graphics Incorporated (SGI) 4D series workstations under the IRIX (SGI UNIX) operating system.

Other, commercially available programs under evaluation at the STB Lab are Inventor<sup>2</sup> and Performer<sup>2</sup>. They serve essentially the same functions as SSM and TDM, respectively. Both programs were developed on SGI IRIX computers under the IRIX operating system, are portable and platform independent. In addition to having similar capabilities to SSM, Inventor includes a library of graphics objects and manipulations based on its object-oriented software library. With its database of scenes, Inventor not only renders, but provides other operations such as picking, printing, event handling, and input/output. Performer, comparable to the TDM program, provides for rapid development and functionality of generated environments. Also, users are given real-time control of the visual system without many of the constraints imposed by traditional image generation systems. The most attractive aspect of programs such as Performer is the availability of numerous and unique special effects that can be incorporated into an environment. Two of particular interest in developing virtual environments are point lights and weather effects. Although many other comparable model and tree development programs exist and are not mentioned here, they will be given similar consideration as they are obtained or made available for evaluation.

Because of the GUI format of these programs, little programming expertise on the part of the developer is necessary to create the 3D graphical models or to establish interrelationships between the models. However, the third VE development activity, environment behavior definition and coding, has generally required the greatest amount of time and programmer expertise. Behaviors, characteristics, and interactions between objects and the user within a VE must be defined via command line C coding prior to rendering the environment scenes. In an effort to simplify this environment behavior definition phase for non-programmers, and to provide for easy access (i.e. common interface) to the model and tree development tools, a graphical interface and development tool has been created. The Visual Interactive Virtual Interface Developer (VIVID) is an X-Windows based system utilizing the Motif Tool Kit and employing a drop-down menu system for user selections. The thrust of this research is to effect rapid development and prototyping of virtual environments.

## VIVID FUNCTIONALITY

At this stage in its development VIVID has three primary functions; program access, behavior editing, and code generation. These functions are accessed via point-and-click activation of drop-down menus. As displayed in Figure 1, the VIVID screen format is common with other graphical user interfaces (GUI) such as those found on Macintosh and PC compatible computers - with some exceptions. Referring to the figure, the FILE, EDIT, VIEW, and HELP menu selections meet standard GUI design specifications. However, the PROPERTIES, CODE, and PROGRAMS choices are specific to the functionality of VIVID mentioned above. A description of each of these menu selections is presented in the following paragraphs. The large window on the left of the depicted screen in Figure 1 contains a list of all objects associated with a specific environment. The environment, and in turn models, is selected as a TDM "tree" via a FILE menu item selection. The large window on the right of the screen lists those objects from the list on the left which have already had properties assigned to them. Adding and deleting models

from this is accomplished with the EDIT menu, and properties are associated with models via the PROPERTIES menu.

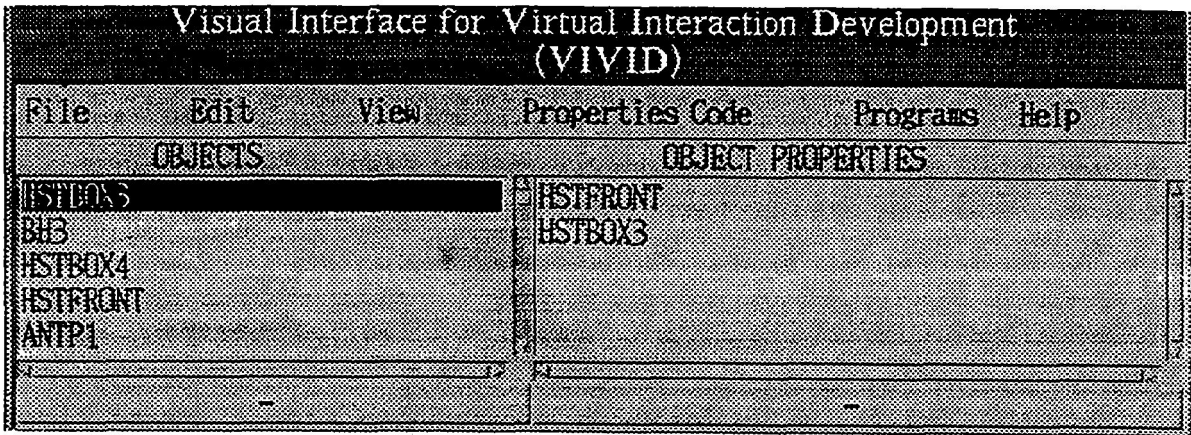


Figure 1 VIVID user interface and top-level menu items.

Initially, the FILE menu item is selected to display file operations commands. The available selections are shown in Figure 2. Currently, the NEW menu item is not functional. At a later date this will be used to create a new executable virtual environment file from a program "template". This template will include the C code necessary to direct the computer software to: load the appropriate model and tree software; render, sinc, and display an environment; and, interface with the data input/output devices. Other function code such as that for interfacing with the sound generating computer will also be included in the template. The LOAD item is presently configured to list for selection, only TDM trees in the configured path. As more development programs are evaluated and used (e.g. Performer), the LOAD item will be expanded with a secondary menu listing. Functionality of the other FILE menu selections; CLOSE, SAVE, and QUIT, is comparable to those of other GUIs.

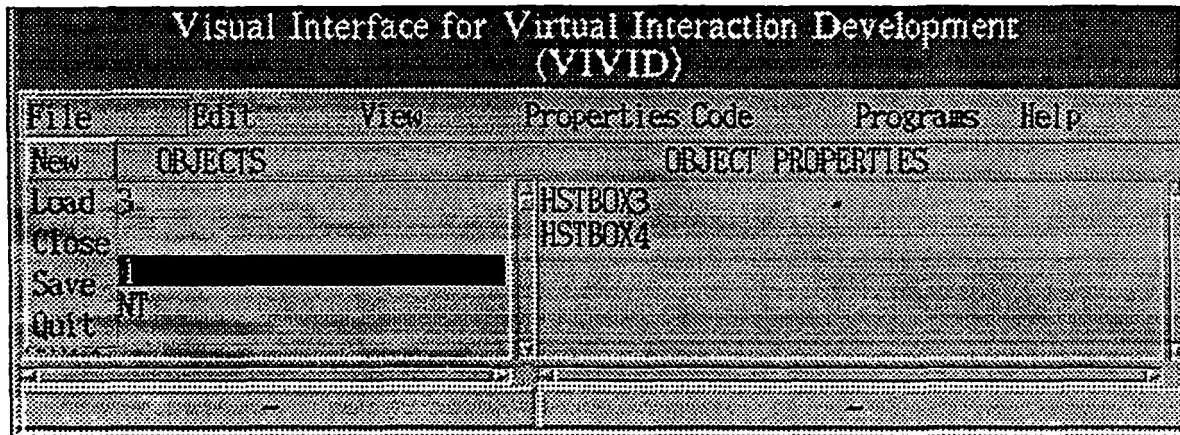


Figure 2 FILE menu item selections.

As shown in the EDIT menu screen, Figure 3, all of the available selections are standard with the exception of TREE and MODEL. The CUT, COPY, PASTE, ADD, and DELETE functions are used to manipulate the items listed in the "Objects" and "Object Properties" windows. An item is first selected (highlighted) with a single mouse click, and the desired operation is initiated. As mentioned, the TREE and MODEL operations are somewhat different from the others. The MODEL option will load an SSM model set and display it's content in the left-hand window. Any models with pre-existing properties will appear in the right-hand window. Selecting TREE will cause a TDM tree - associated models, to be loaded and its corresponding models to appear similarly in the windows. Once models are loaded and listed in either fashion, they can be edited with the desired EDIT menu selection.

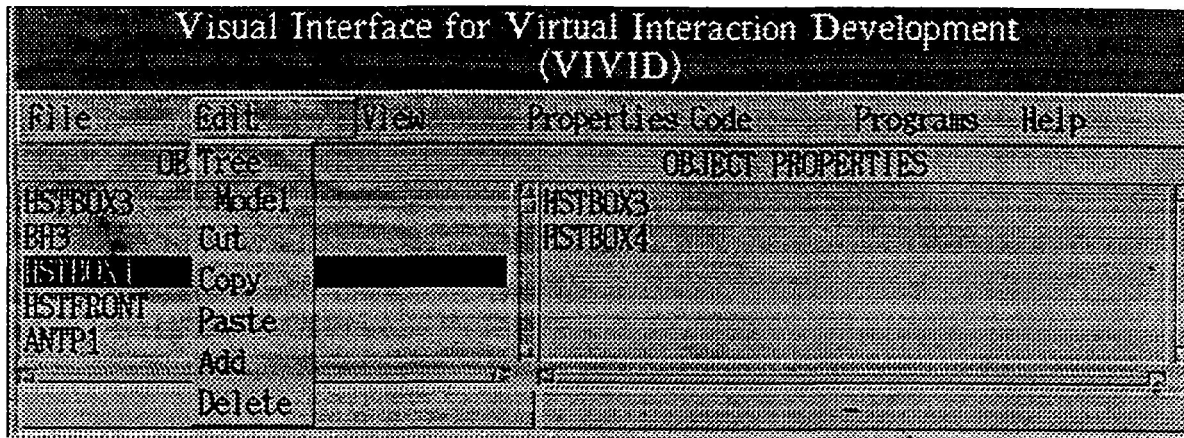


Figure 3 EDIT menu item selections.

To view either a model or a tree that has been loaded via the EDIT menu selections, the TREE or MODEL selections can be activated from the VIEW drop-down menu listing, Figure 4. This function is useful when trying to organize and decide which objects (models) to associate properties with.

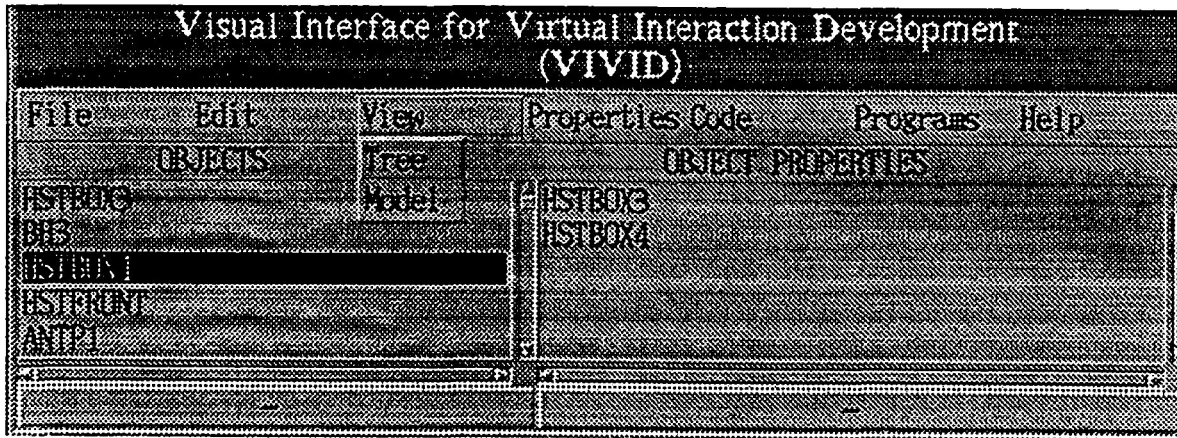


Figure 4 VIEW menu item selections.

Once objects (models) have been added to the Object Properties list the user can select the desired properties to associate with them. These are selected from the drop-down property list of the PROPERTIES menu item, Figure 5. Currently only "grabability", GRASP, and contact sounds, SOUND, are available options on this list. As more behaviors, properties, and "functionalities" are defined and coded for these environments they will be included.

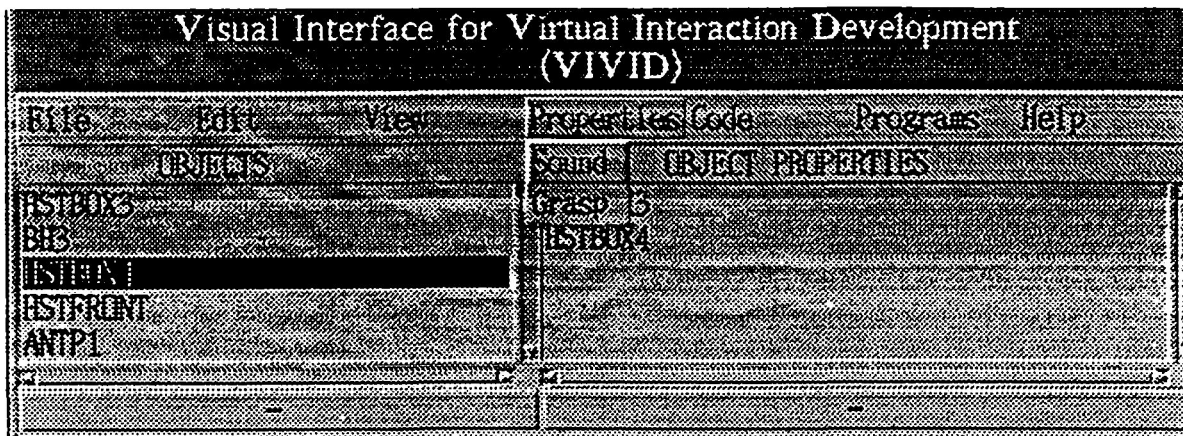


Figure 5 PROPERTIES menu item selections.

After models have been added to the Object Properties list with the EDIT menu, and properties linked to these models with the PROPERTIES menu, it is necessary to update the executable environment program. This is done with the CODE menu items in Figure 6. By simply selecting GENERATE, the C code associated with each of the selected models' properties will be created. After the code is generated it is inserted at the appropriate position in the existing program code. This is the step of the virtual environment development process that takes so much time, but is nearly eliminated with the use of VIVID. The new and existing code must then be compiled into executable form. This is done with the MAKE menu item selection. Finally, the new environment program is tested and verified to be operationally correct (i.e. correct maneuvering, object orientation, and object properties) by selecting the RUN menu item. This will run the program and display it on a screen window alongside the VIVID window. If there are problems or additional modifications needed, the necessary changes can be made by again calling VIVID.

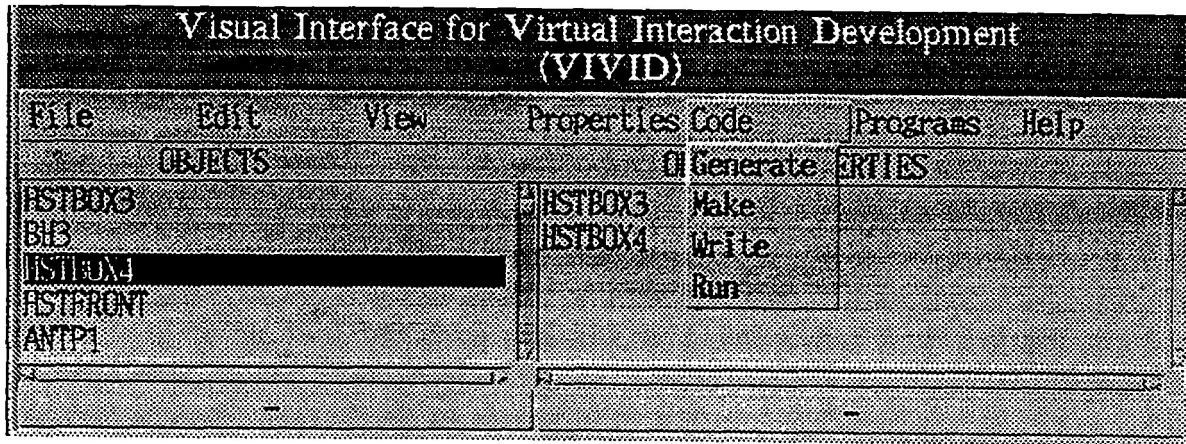


Figure 6 CODE menu item selections.

The PROGRAMS item is basically the interface to other virtual reality development tools. The drop-down options associated with this item can be seen in Figure 7. The SSM selection will run and overlay the Solid Surface Modeler program over the current screen, while selecting TDM will concurrently display it's interface window with the VIVID screen. At this point, either the SSM and TDM program is running and can be used in it's normal mode of operation. SSM can be used to edit or create models, while hierarchical tree relationships can be modified with TDM. VIVID is still in memory and running, but is in the background until SSM or TDM are closed normally. As other virtual environment tools are installed and used, similar single-click access will be provided as menu items.

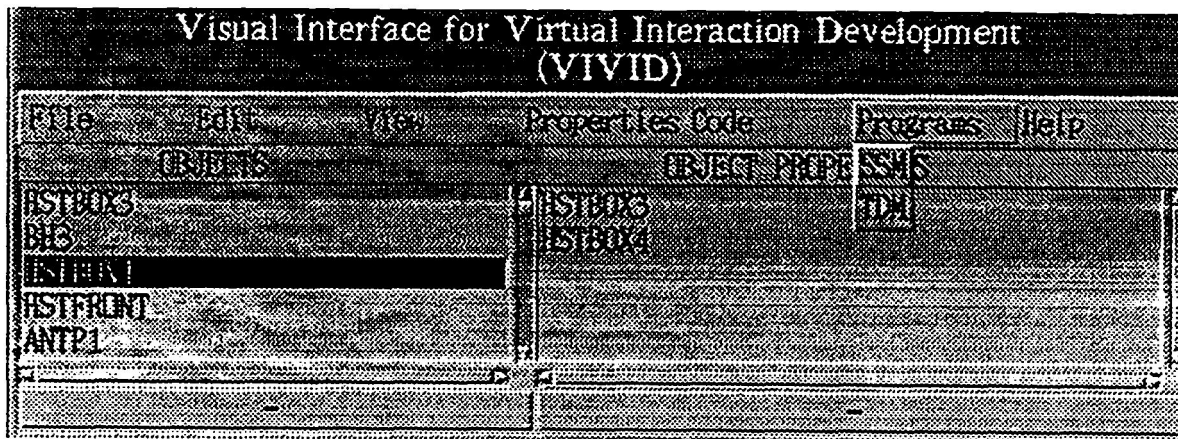


Figure 7 PROGRAMS menu item selections.

## **FUTURE DIRECTIONS**

In conjunction with its existing interfaces and predefined suite of behaviors, future development plans for VIVID include incorporation of multi-user VE capability enhancements, tool expansion, additional behaviors, and interfacing with other graphics library and development programs.

The first user capability to be added is a graphical behavior editor. This consists of a tree diagram of the models in an environment - similar to that found in TDM - and a menu window of available behaviors. Selecting a model will display a pop-up listing of its existing behaviors. These can be added to by "dragging-and-dropping" a new behavior from a menu listing of those available. Another part of this feature will be an editor to add and delete objects within a tree. The second feature to be added is a user capability to create object behaviors. A user can modify their individual menu of available behaviors by adding it to the menu and writing the associated C code. This will be useful as new virtual environments are created and unique behaviors are required. A tool expansion feature will also be incorporated with VIVID. As new peripheral I/O devices are attached to the VE system, associated behaviors can be added to the VIVID menu listing. For example, when a tactile feedback glove is attached at least two new behaviors are possible within a VE; force applied to the user from an object and force applied to an object from the user. Finally, as new, more versatile graphic library and development programs are available VIVID will be modified to interface with them. This will involve data I/O between VIVID and these programs, as well as updating the properties and behaviors available within VIVID for implementation in virtual environments.

- 
- 1. Programs developed by the NASA/JSC Integrated Graphics and Operations Analysis Laboratory (IGOAL).**
  - 2. Programs developed by Silicon Graphics, Incorporated, Mountain View, CA.**