A Virtual Environments Editor for Driving Scenes

Ronald R. Mourant and Sophia-Katerina Marangos
Virtual Environments Laboratory, 334 Snell Engineering Center
Northeastern University, Boston, MA 02115 USA
mourant@coe.neu.edu

ABSTRACT

The goal of this project was to enable the rapid creation of three-dimensional virtual driving environments. We designed and implemented a high-level scene editor that allows a user to construct a driving environment by pasting icons that represent 1) road segments, 2) road signs, 3) trees and 4) buildings. These icons represent two- and three-dimensional objects that have been pre-designed. Icons can be placed in the scene at specific locations (x, y, and z coordinates). The editor includes the capability of a user to "drive" a vehicle using a computer mouse for steering, accelerating and braking. At any time during the process of building a virtual environment, a user may switch to "Run Mode" and inspect the three-dimensional scene by "driving" through it using the mouse. Adjustments and additions can be made to the virtual environment by going back to "Build Mode". Once a user is satisfied with the three-dimensional virtual environment, it can be saved in a file. The file can used with Java3D software that enables the traversing of three-dimensional environments. The process of building virtual environments from pre-designed icons can be applied to many other application areas. It will enable novice computer users to rapidly construct and use three-dimensional virtual environments.

Keywords: virtual environments, editor, driving simulator, Java 3D.

1. INTRODUCTION

The creation of three-dimensional environments for use in driving simulators is not an easy task. Presently, many of these environments are tediously built by specifying polygons and adding textures by hand. Three-dimensional objects can also be build using sophisticated software and then "imported" into the virtual 3D environment. However, this process often involves time-consuming fine-tuning in terms of reducing the number of polygons and eliminating texture-mapping errors. Another problem with this approached is that the placement of the various components in the 3D environment is difficult because it is done via specifying x, y, and z coordinates that are not easy for a user to determine. Patadia [1] eliminated the difficulties in placing 3D objects by developing a visual editor to build driving environments. However, this editor was difficult to use and relied on the Renderware API [2] which is no longer supported.

Papelis [3] uses parameterized tiles for the rapid development of driving visual databases. By allowing any size rectangular tile to be used, he was able to demonstrate how a small number of parameterized tiles could be used to develop complex databases. Recently, Evans [4] has reported the incorporation of pre-fabricated database modules in their system to let end-users build driving environments.

The software system for the easy creation of 3D driving environments that was constructed by Papelis is not available for general use. The system developed by Evans is part of an expensive driving simulator. Both systems lack portability to other platforms and operating systems. Since the virtual environments editor reported in this paper was developed using Java 3D, portability problems should be minimal.

2. DESCRIPTION

The user interface for out Virtual Environments Editor (VEE) is show below in Figure 1.

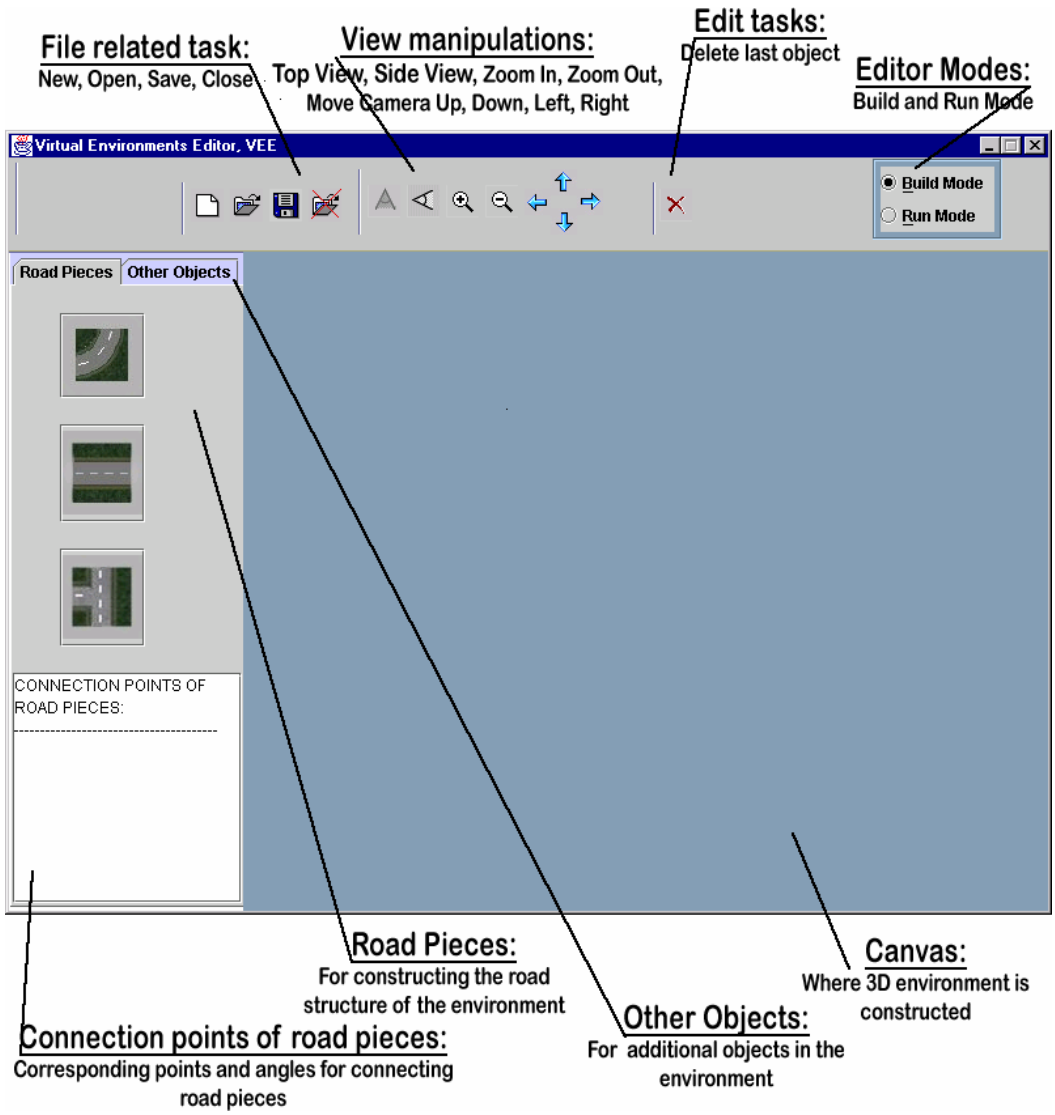

Figure 1. User interface for the Virtual Environments Editor.

The toolbar at the top contains icons for file operations, view manipulations, an editor task, and selection of the editor mode. The left-side panel has tabs for road tiles and other objects. In the bottom left-hand corner is a text area for displaying connection points of road pieces. The large canvas is where the 3D environment is shown. Figure 2 shows the toolbar in more detail.

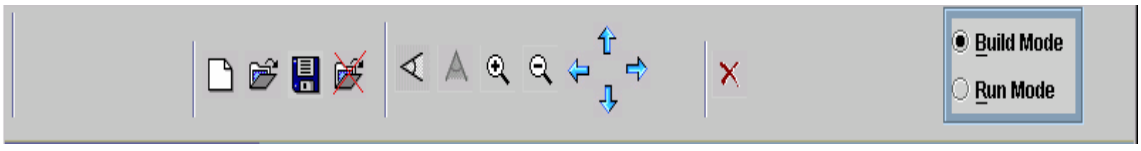

Figure 2. Toolbar of the Virtual Environments Editor

Important capabilities included on the taskbar are the ability to zoom in and zoom out, and to manipulate the position of the camera up, down, left, and right. Thus a user is able to get various views of the 3D driving while it

is being constructed. On the extreme right side of the toolbar are radio buttons that allow the switching back and forth between build mode and run mode. A view of the VEE in run mode is shown in Figure 3.

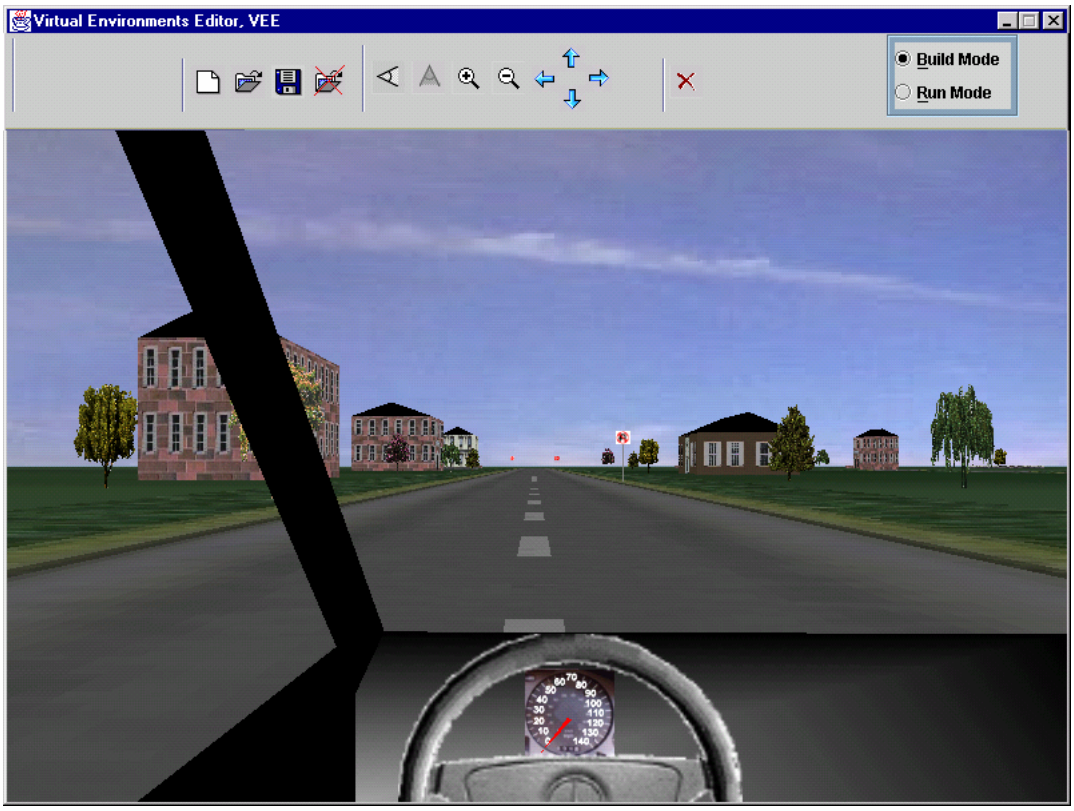

Figure 3. Run Mode of the Virtual Environments Editor.

While in Run Mode, a user may "drive" the vehicle in the virtual environment using the computer's mouse. Pressing on the left mouse button simulates a gas pedal and pressing on the right mouse button simulates a brake pedal. Moving the mouse horizontally (left and right) simulates a steering wheel. This examination of the environment while in Run Mode is very valuable in term of road design and placement of objects.

Figure 4 shows the results of pressing both the Road Pieces and Other Objects tabs.

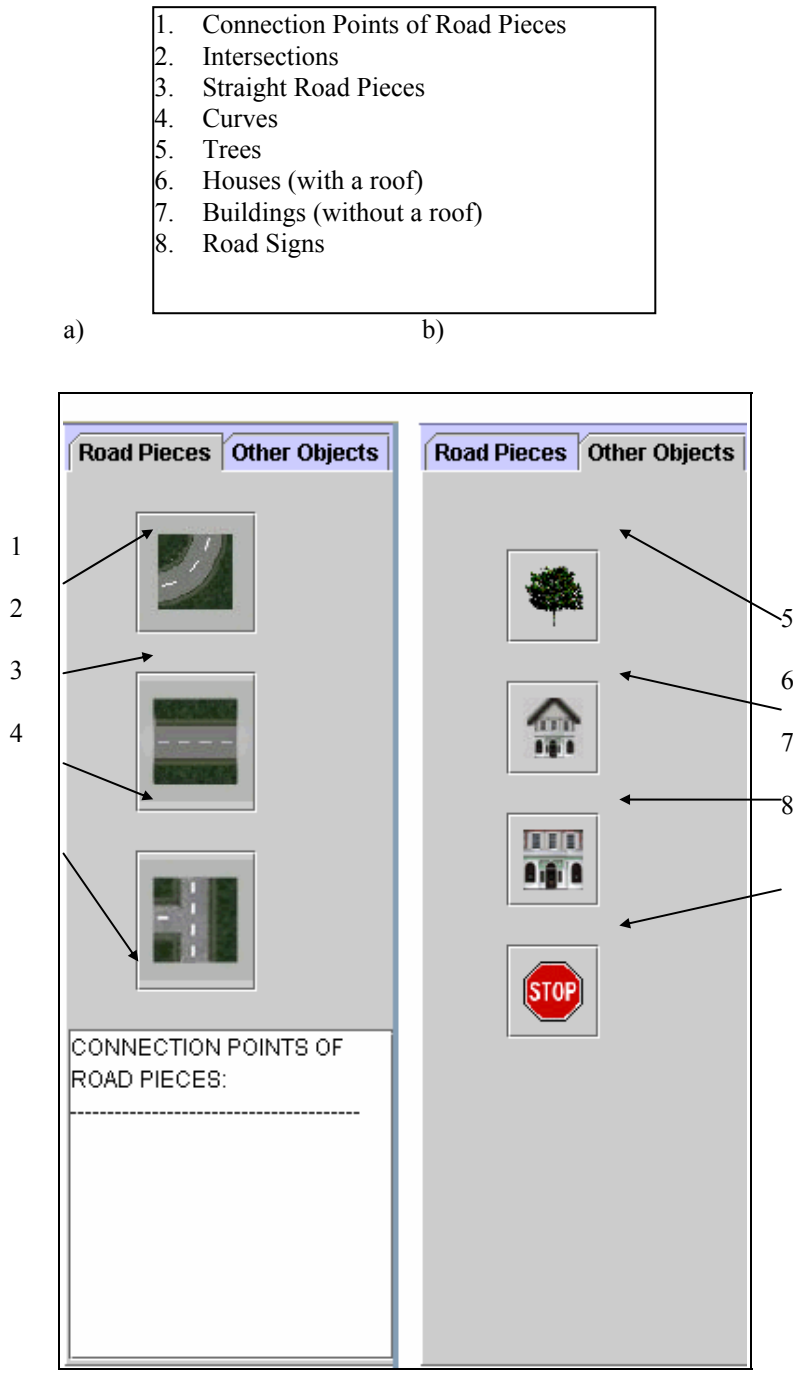

Figure 4. Road Pieces and Other Objects in the VEE.

Currently the road tiles are two-lane roads that are intersections, straight sections, and curves. Parameters of road tiles include starting position (x, y, z), angle, and length. Possible starting positions for road tiles are shown as connection points in the text area at the bottom left of the screen.

Clicking on some tiles, brings up a dialog box that allows the choice of different objects types of that object. For example, clicking on the stop sign tile, brings up the dialog box shown in Figure 5.

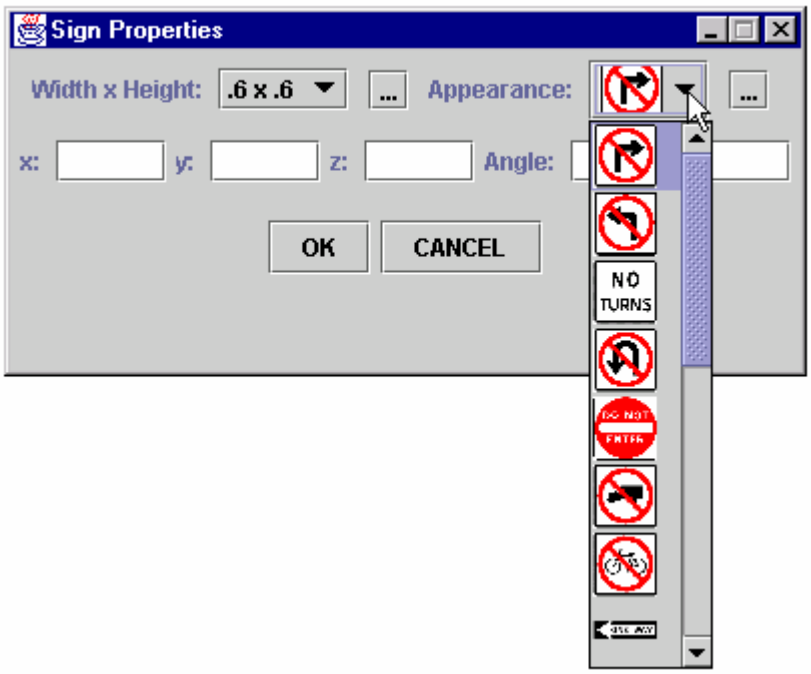

Figure 5. Dialog box for sign tiles.

Sign tiles are shown in the drop down choice menu. The available widths and heights of a sign are also displayed via a drop down choice menu. By pressing on the buttons with an ellipsis, a user may add more choices to widths and heights and more sign tiles. Thus, the VEE may easily accommodate most road signs. Each combination of a sign tile and a width and height, represents a bitmap file of the sign image.

When displaying signs in the virtual driving environment a MIP-mapping algorithm is used as suggested by Bouvier [5].

3. DISCUSSION AND FUTURE ENHANCEMENTS

We have shown that it is possible to rapidly develop virtual driving environments using low-cost desktop computers. Some driving environments that take weeks and months to model, can now be built in a few hours. Further, major improvements in graphics cards for PCs have resulted in very fast frame rates making possible such features as a run mode in the VEE. Yasuda, et. al. [6] have built a editor for the development of driving environments for high-performance graphic workstations. Their system is not titled based but does allow the construction of sophisticated road networks. Some of their technology can be incorporated into the VEE.

Some future enhancements include the following:

Drag and Drop: A drag and drop capability could be added to the editor. This will enable the user to place objects without having to input their position. Filling the respective dialog box for each dragged object is still necessary because the user will still need to input information such as angle and length.

Picking: A picking capability could also be added. This will enable the user to pick objects in the "Build Mode" and move them around for easy repositioning. When an object is picked, the object should be highlighted so the user can distinguish what objects are picked.

Autonomous Vehicles: The capability of adding other moving traffic to the driving environments is desirable. A car icon would be added to the other objects tab for this capability. When this icon is pressed, a Car Properties dialog box would appear to ask the user to fill in appropriate information about the car. Some of the information will be on the type and color of the car, the path the car is to follow, and the vehicle velocity. The car

would appear idle at its starting position when the VEE is in "Build Mode". It would start moving when the mode is switched to "Run Mode".

Traffic Signals: Traffic signals can also be added. A traffic light icon can be added to the other objects tab. When this icon is pressed, a "Traffic Light Properties" dialog box would appear to ask the user to fill in appropriate information about the light. Part of the information needed would be the position at which the traffic light should be placed in the environment. The traffic light would appear in the environment when the VEE is in "Build Mode" and start functioning i.e. switching from red to yellow to green at constant intervals, when the mode is switched to "Run Mode".

Ground Elevation: As of now, the VEE is capable of essentially building any type of environment. All objects can be placed at an elevation, including road objects, but the capability of driving on elevated roads is not implemented. Moreover, elevated roads should be constructed with a certain rate of incline according to road regulations. This would be a good enhancement to the editor because it would make the editor capable of creating even more realistic driving environments. The study of driving on hilly, curvy roads can easily be done when using the editor after such an enhancement.

More Road Options: All the road-related dialogs, the straight road, the intersections and the curves dialogs, can be changed to accommodate a selection of road pieces with different looks. At the moment, all road objects contain grass on the side of the road, so for instance, a road with a side walk, or a road with a solid yellow line, can be one of the possible choices that the user can choose from, using a pull down menu. This functionality will be just like the one provided for trees, where the tree dialog box provides the user with a list of different types of trees he/she can chose from (refer to Figure 12). Another property to add is the ability to state how many lanes the road should have. At the moment, the VEE only creates two-way single lane roads.

Predefined Environment Tiles: Predefined environment tiles are areas of the environment considered to make up one object. A simple tile can contain many road objects, houses along the road, road signs, and trees, etc. Tiles can greatly simplify construction of environments, because the user does not have to define the position, size, angle, etc. of many objects along the road. A tab called "Predefined Environment Tiles" could be added to go with the "Road Pieces" tab and the "Other Objects" tab. Icons representing different tiles can be added to the "Predefined Environment Tiles" tab. These tiles can represent different types of environments such as farmland, mountains, forests, cities, neighborhoods and hilly land. When one of these icons is pressed, a properties dialog box would appear to ask the user to fill in position, length, and angle for the entire tile. Clicking on OK will add the tile to the environment. Its road connection points would still show up in a text area of the tab, to enable the user to connect tiles together.

4. REFERENCES

[1] J. Patadia, "A Modeling Language For Creating Driving Environments", Unpublished Master's Thesis, Northeastern University, December, 1998.

[2] R.F. Ferraro, **Learn 3D Graphics Programming on the PC**, Addison-Wesley, Reading, MA, 1996.

[3] Y. Papelis, "Rapid Development of Domain Specific Correlated Databases Using Parameterized Tiles", **Proceedings of Image'98 Conference**, 1998.

[4] D. F. Evans, "Fast and Beautiful: New Approaches for Creating Virtual Environments", **Proceedings of Image'2000 Conference**, 2000.

[5] J.D.Bouvier, "Getting Started with the Java 3D API - A Tutorial for Beginners." http://www.java.sun.com/products/java-media/3D/collateral/, Sun Microsystems, Inc. 2000.

[6] T. Yasuda, T. Suzuki, S. Yokoi, and J. Toriwaki, "Virtual Environment Construction for Driving Simultor", **Proceedings of the IEEE International Workshop on Robot and Human Communication**, p. 50-55, 1994.