Implementation of Virtual Environment using VIRTOOLS

Mustafa Agil Muhamad Balbed, Nazrita Ibrahim*, Azmi Mohd Yusof  
College of IT, Universiti Tenaga Nasional, Malaysia  
Balbed@uniten.edu.my, Nazrita@uniten.edu.my, Azmiy@uniten.edu.my

Abstract
VRML is one of the most common and popular virtual reality development tools available in the market. This low end product is widely used in school or college. High end product such as Sense8, EON, Cartia and many others have been in the market for quite number of years. These applications have been popularly used in the big industries such as medical, automobile and infrastructure design. However, for our purpose of environment visualization and immersion, Virtools development kit was used. The main focus of this paper is to describe the approach used in producing an immersive and interactive 3D environment used to treat acrophobia. The 3D environment consists of a busy city surrounded by tall buildings. The advantages and benefit of using Virtools as Virtual Reality development tool is then explained.

Keywords: 3D Environment, Virtual Reality, Immersive, Virtools.

1. Introduction
Virtual Reality (VR) is a term referring to a combination of high-speed computers, advanced programming techniques and interactive devices that allow a user to interact with a computer-simulated environment [1]. Most current Virtual Environments (VE) are primarily visual experiences, displayed either on a computer screen or through special or stereoscopic displays, but some simulations include additional sensory information, such as sound through speakers or headphones [2]. Virtual Reality has long been used in industries such as automotive, medical and infrastructure design [3][4]. In any virtual reality environment, the truly immersive environment can only be created when the whole images and environment are photorealistic.

Another very important factor toward a successful immersive virtual environment is the availability of real time interaction and rendering. For virtual development developer, it is necessary to quickly generate prototypes and to test different interaction techniques and graphical representations, especially if major parts of the final environment are not yet available. This requires a visual authoring environment that allows the developers to switch between design elements, provides a repository of advanced interaction techniques, and easily allows generating new elements. However, it is difficult to have a development tool that provides all the function in one package: modeling, animation, video and audio editing, interaction and real time rendering in the immersive environment.

This project is about developing a virtual environment which could simulate the presence of height. The environment will then be used to treat patient suffering from acrophobia (fear of height). In this paper, we concentrate on outlining the steps taken in developing the environment, using Virtools as the main engine. At the end of the development, we are convinced that Virtools is a good and reliable development tool towards implementing a truly immersive virtual and interactive environment. We have no intention of comparing Virtools with well known packages (tools) in term of speed and quality.

2. System Architecture
The architecture of the system is shown in Figure 1. Firstly, the environment and all the associated objects are modeled using MAYA. The objects are the buildings, the road, the terrain, the car, the elevator and the helicopter. The objects are then being exported to Virtools, and arranged based on sketch shown by Figure 2.
3. Environment Modeling Using MAYA

The following objects have been modeled using MAYA:

i. The buildings
ii. The terrain
iii. The Scaffolding elevator
iv. The trees.

The following objects have been taken from the ready-made 2D internet objects and modified using 3dsMax:

i. The cars
ii. The helicopter

3.1 Modeling the buildings

There are seven different buildings been created. The main building is created by extruding the polygon cube and applying the Lambert material with a little bit of transparency effect. The model is constructed using repetitive method where we created the shape of the first floor, and stacking the copy of the first floor to make up the rest of the floor. The pictures of the main building can be seen in Figure 3.

The following are the rest of the buildings which fill up the environment. All the six buildings are created from polygon, with ‘bevel’ effect being applied to each one of them. Texture mapping and bump mapping are applied to every building. The screenshot of the buildings can be seen in Figure 4.
3.2 Modeling the Terrain

The terrain is used as a ground of all the elements and is placed below the road. The terrain is created by sculpting plane material so that it will look like hill rather than flat. In addition, the edited picture of grass is used as a texture map of the terrain. The snapshot of the terrain is shown in Figure 5.

![Figure 5: The Terrain](image)

3.3 Modeling the Scaffolding Elevator

The scaffolding elevator is created by using polygon. The based and the pillar of the elevator are created using cube, while the railing are done using cylinder. Metal image is then texture-mapped to the elevator. The snapshot of the elevator is shown in Figure 6.

![Figure 6: The Scaffolding Elevator.](image)

3.4 Modeling the Trees

The trees are used to add the realism to the environment. The trees are placed in every corner of the road and in front of the each building. The trees are taken from the built-in model provided in Maya. The picture of the trees can be seen in Figure 7.

![Figure 7: The Trees](image)

3.5 Modeling the Cars and Helicopter

Both the cars and helicopter are taken from the ready-made model available from the internet. The models are then modified in terms of the shapes and colours. The snapshots of the car and helicopter are shown in Figure 8 and Figure 9.

![Figure 8: The Car](image)

![Figure 9: The Helicopter](image)

Once the modeling phase has been completed, all the objects are exported to Virtools. There are five main actions that need to be done in Virtools, such as:

i. Objects arrangement
ii. Animation of the cars
iii. Adding the stereotype sound to increase reality
iv. Incorporate interaction
v. Integration with the HMD

4.1 Object location and arrangement in Virtools

All the elements imported from Maya must be arranged carefully. Resizing and locating the elements in the right place need also be done in order to make sure that the environment relates properly to human size. The arrangement of the environment can be seen in Figure 10.

![Figure 10: The arrangement of the environment.](image)

4.2 Animating the cars

The cars need to be animated to mimic the real world scenario. The road is designed so that the traffic can flow in both directions. To animate the car, we use path animation. We create a path for the car to follow. Since the road is two-ways road, 2 path are set up, one for each direction. The position of the cars with the visible path can be seen in Figure 11, while Figure 12 shows the bird eye view of the city road.

![Figure 11: Position of the cars.](image)

![Figure 12: Bird eye view of the cars.](image)

4.3 Adding the stereotype sound to mimic reality

In the effort of making the environment to be more realistic, we include the stereotype sound of each object in the environment such as the helicopter, the wind, the elevator and the car passing by. We attached each sound to the related object. For example, the elevator sound was attached to the elevator; the car sound is attached to the car and the sound of helicopter to the helicopter itself.

By attaching the sound to the object, it was easier for us to simulate distance in the virtual environment. The further away the object to the character that we are controlling, the softer the sound will be. The nearer we are to the object, the louder the sound will be. By using stereotype sound, the sound could be listened dynamically rather than static. For example, when the car moves on the left hand side of the listener, the left speaker will become louder compared to the right speaker.
However, the wind sound was not attached to any object but was set as background. We configured the sound of the wind so that it could only be heard when the elevator is raised to a certain height. The sound was disabled when the elevator is on the ground. Figure 13 shows the snapshot of the sound set up in Virtools.

![Figure 13: The block diagram of sound.](image)

4.4 Incorporating interaction

To replicate the user movement in the environment, we created a character to represent the user in the virtual environment. To move around the environment, the user basically will need to control the virtual character. As the character move around it will automatically update what user see inside the virtual environment. The view that the user sees was attached to the character through the use of camera.

We applied collision detection in the environment to avoid the character from walking through the objects inside the environment. The implementation of the constraint is applied using one of the Virtools building block or modules called Object Slider. The usage of object slider will allow us to simulate the collision detection. When the character controlled by the user collide with the object in the environment, the character will slides along the object surfaces.

Character Controller Building Block is used to add animations (e.g. move forward, backward, jump, and run) to the character. Each of the animation is assigned to the arrow key on the keyboard using Keyboard Mapper Block. The Keyboard Mapper allows us to create a controller for the user. The limitation of hardware possession forcing us to rely on the keyboard as the interaction device in interacting with the environment. The Character Keep-on-floor Building Block is used to keep character on a user-defined floor which in our case is the road. Fig. 14 shows the detail set up of the Character Controller and Keeps-on-floor building block.

![Fig. 14: Setup of character and object slider](image)

4.5 Integration with Head Mounted Device (HMD)

In our case, we were using a stereoscopic 5DT Head Mounted Display with the following specification: 800-26 Resolution 800x600 per display, 26° Diagonal Field-of-View, LCOS Display Technology and Sennheiser HD25 (16Hz - 22kHz) Headphones.
The environment runs on Pentium 4 3GHz Graphic Workstation, 2 GB RAM using NVIDIA graphic card (Quadro FX1400, 128 Mb) as shown in Figure 16. There are some settings [5] that need to be made to the display before we can connect the HMD to the PC.

First, we need to set the 3D display setting to ‘clone’ so that the environment is projected to both the monitor and to the HMD screen. Second, we need to set the resolution of the virtual world to 800 x 600.

Finally, the HMD is connected to the PC using the VGA port. Figure 17 shows the details of how to connect the HMD to the PC.

Figure 16: 5DT HMD.

Figure 17: The installation of HMD.

5. Discussion

Many virtual reality systems are developed by using either programming language such as OpenGL or modeling language such as VRML. The problems in using these languages are that everything must be specified using the program, including the positioning of the objects in the environment and the interactivity between the users and the environment.

Positioning using program can be such a headache since everything needs to be specified using coordinate. The developers need to be very well verse with the environment coordinate system in order to make sure that an object is placed at the correct position. This can be quite a tedious job to do since over times we tend to move things around.

Virtools simplify the development process since it incorporates necessary interface to help the developer expedite the work. The preview of the environment and all the objects is similar to any other 3D modeling software. Each object can be positioned and manipulated by using the ‘drag and drop’ technique. We can also create light and camera objects in Virtools.

The existing library is packed with useful built-in function. The function in Virtools is represented using building block. Each building block is like a pre-written program, and what we need to do is to feed the block with the necessary input. For example, there is a built-in character movement in Virtools such as walk, run, jump or idle. We could also specify the camera view whether it is a first person view or a third person view or simply a perspective view.

Figure 18 and 19 show snapshots of the usage of building blocks in Virtools. One building block works as a function in structural programming (like C), and each building block could receive input parameters and sent out output parameter [6]. You can have one building block, connected to another building block through parameters, and building block within another building block (function within another function). Virtools incorporates what we call “Visual Programming”. You see the program visually as in chunk of block.

Figure 18: Example of a graph of behavior building block in Virtools

Figure 19: Example of Building Block with parameter coming in and out
Virtools also comes with built-in building blocks that allow promising integration with HMD and Data Glove.

One of the drawbacks of using Virtools is its steep learning curve. It could take quite sometimes to understand the way things work in Virtools. References in terms of books and online help are limited.

Despite of the drawback, Virtools is still a better tool to use when it comes to developing the virtual interaction.

6. References


