3D OBJECT IMPLEMENTATION ON BICYCLING AT UI VIRTUAL REALITY APPLICATION BASED ON 3D-GAMESTUDIO

Rini Fitri Badi, Anna Gunty, Citra Purwansari, Prima Dewi Purwansari
Department of Electrical Engineering Faculty of Engineering
University of Indonesia
Kampus UI, Depok 16424, Indonesia
Email: rini, anna.gunty, citra.purwansari, primadelp@eng.ui.ac.id

Abstract — This paper reviews 3D computer technology and our experience in creating a virtual bicycling environment at the University of Indonesia (UI) campus. We explain the implementation of the VR environment using 3D-Games Studio and our experience in viewing the result with VR device, i.e., 3D E-Dimensional wireless goggles.

In this work, we present the real world elements with the graphics that imitate the real world such as bicycle movement, camera perceptions, and object collisions handling to other entities such as wall, tree, or building. The environment includes bicycle track, trees and obstacles in the pathway.

This application is subsequently tested by the users in terms of the general object conditions, user's response to the virtual reality environment and the future development.

I. INTRODUCTION

The advancement in technology kept driving more development in many areas. There are many sophisticated games and applications exist during the last 20 years. 3D computer technology grows rapidly and provides some alternatives of 3D software applications to help making our imagination come true. 3D technology application or virtual reality enables people to feel their presence in a specific virtual world. In 2007, a 40 kilometers bicycle track has been constructed on the grounds of the University of Indonesia (UI). More than 1000 bicyclists have been provided by the University to be used freely by the student and staff to encourage the use of healthy and affordable transportation facility. The bicycle track was built according to the green campus and world class eco-campus concept. This program shows UI commitment to address global warming issue.

We proposed to create an application that can stimulate students to ride bicycle inside campus. It should simulate the real bicycle track environment. Students should be ensured that in addition to healthy lifestyle, cycling at the university's environment is also safe and fun, by means of having the experience from riding the VR program. The main object in this application is a bicycle that moves on UI bicycle track. Imitating the real bicycle, in this application we provide some movement method for the bicycle such as forward, backward, turn left and right. To complete the real effect, the application is added with camera perceptions. The main perception is first person that can see the environment as if one ride the bicycle. Another perception is a third person perception which acts as a close camera that orbits the bicycle.

The purpose of the application is to create a complete bicycle object and environment with some behavior which have a direct and indirect perspective camera, built using 3D GameStudio. This tool has three editors, i.e. World Editor, Script Editor, and Model Editor. The application is evaluated and viewed using E-Dimensional 3D wireless glasses for PC.

With VR we can see that the view in the glasses is more real, as if it is really in front of us. The glasses used in this experiment are wireless glasses which receive a signal to synchronize the glasses with the 3D image on the monitor. The 3D goggles in this experiment used the same concept with the equipment the 3D movie in the theatre.

II. LITERATURE REVIEW

A. Virtual Reality

Virtual reality (VR) is a technology which allows a user to interact with a computer-simulated environment, be it a real or imagined one. Users could wear many devices that can translate the movement to be used for manipulating virtual object [2]. Most current virtual reality environments are primarily virtual 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 [1].

Users can interact with a virtual environment or virtual artifact (VA) either through the use of standard input devices such as a keyboard and mouse, or through multidimensional devices such as a virtual glove, the Polhemus beeem scope, and multi-directional treadmill [1].

B. Stereoscopic Equipment

One type of stereoscopic or binocular vision is 3D glasses. E-Dimensional 3D wireless glasses for PC are examples of these stereo-3D-devices. These glasses will help our eyes to see the real 3D vision effect of the simulations that will be viewed. These wireless glasses are not only having dangle as connector but also have one additional component, the infrared transmitter.

Stereoscopy is useful in viewing images rendered from large multi-dimensional data sets such as those produced by experimental data. The three-dimensional depth information can be reconstructed from two images using a computer corresponds the pixels in the left and right images [7].
Dongle is a triangular box with a VOA cable coming out of one side. The dongle will control the glasses as well as synchronize the glasses with the monitor. The dongle has three connections on the back, one for the monitor in the middle, the larger circular one for the wireless transmitter and the other smaller silver connection for the wired glasses to plug in directly. Currently the dongle only supports a VOA connection, not DVI [8].

2. Wireless Glasses
The glasses work as a filter for what we see on monitor. We will know something 3D is being displayed when the screen looks blurry to the naked eye without the glasses on.

3. Transmitter
The transmitter is used to transmit signal produced by display device. The glasses for the wireless glasses plug directly in to the dongle. The transmitter is placed on the top of the receiver.

4. Monitor
There are two kinds of monitor that mostly used, i.e., CRT and LCD monitor. A CRT monitor is a thick and heavy monitor with a glass front part. An LCD monitor is a thin monitor that has a plastic-like material for the screen and can typically be lifted easily with one hand.

C. PR Applications
3D computer graphics software refers to programs used to create 3D computer-generated imagery. Most of the 3D packages have plug-in-oriented architecture costing less or hundreds of thousands of dollars are often used by studios [9], such as 3ds Max, Blender, Cinema 4D, LightWave 3D. For example, 3ds Max (Autodesk), which is used in the video game industry for developing models and creating cinematic scenes. Blender (Blender Foundation) is a free, open-source, 3D studio for animation, modeling, rendering, and texturing offering a feature set comparable to high-end and mid-range 3D animation studios. Cinema 4D (MAXON) is claimed to be user-friendly, and is designed with the low-technical user in mind. LightWave 3D (NewTek) was originally bundled as part of the Video Toaster package and entered the market as a low-cost way for TV production companies to create quality graphics for their programming. Maya (Autodesk) has a high learning curve but has developed over the years into an application platform in and of itself through extendibility via its MEL programming language. Softimage XSI (Avid) has additional features and integrates with mental ray rendering.

In 3D programming there are popular terms, i.e., 3D engine, 3D language, and 3D authoring system. A 3D engine is a library of 3D graphics functions. Many 3D engines are available on the Internet, some (even good ones) are free, some for commercial use. 3D engines require programming with an external development system, usually Microsoft Visual C++, Java, etc. Running a 3D application requires a 3D engine. Thus a maximum flexibility, especially since the programmer usually have access to the engine's source code. However it also requires a maximum of effort and time to be invested before something is moving in a game or applications [4].

An easier approach is offered by a 3D Language, which is a scripting language specially designed for 3D games. Each language do not offer the flexibility of a 3D engine, but avoid a lot of problems related to "real" programming. Many 3D languages are BASIC, C or Java-based languages, which are better suited for large program or complex projects [4].

The next step is to use a game or 3D application is an authoring system, which has its own 3D engine and a virtual editor. Only simple applications can be created without any programming. This authoring system, normally also provide a scripting language for programming or customizing the game. Without an authoring system, an application can be completely developed without the need to understand the source code and library functions of a 3D engine [4].

There are various tools for special applications. Programmers can choose anything due to their needs. For the next part, we will be using about 3D Game Studio, a 3D software that is used to make the simulation in this project.

D. 3D Game Studio
3D Game Studio, often known as GameStudio or 3DSS for short, is a 3D computer game development system which allows users to create 3D games and other virtual reality applications, and publish them royalty-free. It comes with a model/terrain editor, a level editor, and a script editor and debugger, and also contains a texture and level collection [5]. GameStudio can be used to create 3D and 2D games, simulations, or other multimedia applications. A 3D game normally consists of one or several virtual environments - Levels. A level is built from geometric blocks, irregular terrain, variable entities, as well as special items like light and sound sources and action paths. Jason patterns, called Textures, are put onto the surface of every block. Entities can be of various shapes, and their material can have certain properties, like being fluid or solid. Entities can be simple shapes, animated models, or sub-levels. Anything that is moving, like a door or a monster, is under control of a program or script. Scripts are also responsible for the user interface, like dynamic lights, those, particles, fog, or shaders [6].

3D Game Studio needs Model Editor (MIE) or World Editor (WIE), and we can use 3D party games if needed. 3D Game Studio supports many of the popular formats. For example, we use all variables, functions and actions and go to any of them with one click of the button [7]. In 3D Game Studio, there is a large library of 1000 professional textures, building parts, furniture, vehicles, weapons and actions are included [1]. For complex and custom games or other applications, we need to use either the integrated scripting language C++ or an external development systems such as Visual C++ or Delphi [1].

Many 3D game systems use scripting languages for controlling objects or actions. The more things move in the game, the more script languages have to be executed per second. Most script languages are interpreted. This means
that instructions are first translated into an intermediate byte code. The processor interprets the code byte by byte at run time, which causes a slow execution and affects the frame rate. A script compiler translates the language into real machine code—the native language of the processor. Compiled scripts run up to 10 times faster than interpreted scripts, and do not influence the frame rate even in huge games with hundreds of simultaneously moving objects [4].

S3D is like a general plain text editor with a complete and debug tool. The script editor is used to program in Lite-C or C-Script, a scripting language similar to C used in previous generations but supported for compatibility.

1. Lite-C [7,8]

Lite-C is a programming language dedicated to the creation of multimedia applications and computer games, with syntax similar to the C programming language. The difference to C is the native support of multimedia objects like sounds, images, mouse movements, user interface elements, 2D and 3D models, terrain, game levels, collision detection and rigid body physics. The intention is to achieve quick results using only a few lines of Lite-C code [7].

Lite-C supports PHX, DDS, X, OBJ, ARIE, MAP, MDL, MD2, FX, BMP, PCX, and Text Prototype *File* format. It has the following features:

- Extended C syntax, easy and transparent multitasking.
- Compile on the fly to machine code executable.
- Easy integration of external APIs (OpenGL, DirectX) with DLL or COM interface.
- Powerful AIB (Adaptive Binary Tree) rendering engine.
- 2D and 3D sprites and models, vertex and bone animation.
- Programmable particle effect generation.
- Rigid body physics and collision engine.
- Layered sky system with sky cubes, sky dome, clouds and background images.
- Built-in texture, matrix, physics and collision functions.
- GUI objects with button, slider, gauge elements, transparent and bitmap fonts.
- Supports all DirectX 9 functions.
- Plays functions for sound, music and movie files and CD tracks.
- Syntax-highlighting script editor with single-step debugger.

2. WorldEditor (WED)

WED is the editor for creating the virtual worlds, also called levels. It also serves as the 'control center' where levels, models, and scripts will be linked together to the final game [6].

With WED, the various objects can be arranged, actions can be assigned to objects (also known as entities), which are defined through scripts, textures can be assigned to level geometry, and levels can be built using library space partitioning tree (or BSP for short) techniques.

WED is the main program of GameStudio where all parts of the game or application (programming, 3D graphics, levels) can be managed [5]. Fig. 1 shows the overall layout of WED.

Fig. 1 WED Layout

The layout for WED is fairly simple. The main part, the central right section, is where most of the editing is done. There are three graphs and a 3D view. The graphs are split into multiples of 128 and further split into multiples of 16 to help with snapping and spacing. The graphs will automatically resize when user zooms out a bit and in steps of multiples of 8 (1024 and 128 then 512 and 64).

The left central section lists objects in user level, textures, and many other things. The top is the tool bar which allows user to manipulate objects, add new objects (such as entities, scenes, and lights) and build levels.

Entering a position, assigning an action to an entity or redetermination of the texture to the individual side of block can be done by right-clicking on something and choosing properties [5].

3. Model Editor (MMD)

The model editor (or MMD for short), provides the capabilities of designing models, and sometimes used to make levels with. Models can be made of anything ranging from simple boxes, to human models or complex environments like an entire city. Models are made of meshes, a group of vertices and triangles (often called "polygons") put together to form a shape, a skeleton of bones for animation; one or several textures for the skin; and effect files (fx) for shadows [5]. Fig. 2 shows the overall layout of MMD.

Similar to WED, MMD's general layout is with those graphs and 3D view. They are arranged in the same way as WED. However, by default, MMD does not have the grids, but it can be configured that way. MMD has a skin editor, which allows the model to be textured. The skin editor has a completely different layout. The texture is shown on the left side and the model is shown on the right with tool bars sur-
can directly save their models in GameStudio format by means of a plugin. Plugins for creating GameStudio models with MAX or Maya can be downloaded from the GameStudio site [6].

b. **Sprites**

A sprite (also called billboard) is a 2D 3D object that can serve several purposes. It can be placed at a wall or window. It can stand upright in the landscape like a billboard, or can behave like pseudo-3D by always facing the camera.

Sprites are stored in external PNG, BMP, TGA, or DDS files and can be created using a standard paint program, like Gimp, PaintShop Pro® or Adobe Photoshop®. TGA or DDS files can contain an alpha channel which gives a transparency value for each single pixel. PNG, BMP, or TGA files are animated. DDS files can contain normal maps for better quality and faster rendering. Sprite entities are rendered faster than map or model entities, and can be used for explosions, lights, flames, trees, grass, or fire like [6].

c. **Sublevel (Map Entities)**

Map entities are simply a small compiled level, stored in an external WMB file. Map entities can be used for level parts which move as a whole, like doors, platforms, or vehicles. Because they are nothing else than compiled maps, it can be created using WED.

Textures and shadow maps of the level and of map entities are pre-allocated in video memory at loading time, in order to maintain smooth game play. Textures of all other entity types are only allocated when the entity becomes visible [6].

d. **Terrain**

Terrain consists of one or several terrains mapped onto a rectangular grid of height values. It is stored in an external BMP file. As the name says, terrain entities can be used for level parts that are irregular terrain. They can be created with M2 or imported from RAW height maps or BMP or PNG height map images created with terrain builder programs. Terrain can not be animated, rotated, moved, or scaled; however it can be deformed in real time by the engine. The terrain is projected vertically, for that reason the terrain in slopes will look lopsided.

The engine supports two types of terrain, unchunked or C-chunked. Which type a terrain has is determined by the terrain chunk variable or by a file name ending with "_n" (e.g. "americana_n.bmp"). Unchunked terrain is rendered just like a model, and must not exceed a size of 128x128 vertices. The engine's sophisticated terrain rendering algorithms unfold its power with chunked terrain [6].

III. **Device and Architecture**

In this work we design a VR environment and make a 3D bicycle object with some moving behaviors such as forward, backward, turn to the left and to the right, and stop. The addition parts are 1st person camera that act as a direct camera from the object and 3rd person camera that orbit or act as chase camera.
A. Architectural Design

Architectural design will be shown using Unified Modeling system (using 3 diagrams). The Use Case diagram is the diagram that shows the flow of activities of the system. This activity diagram is seen on Fig. 3. Use-case is a technique based on scenario. It is an activity that is needed to be done in the object method. Use-case generally has a UML notation base feature to describe the system model. In the simplest form, use-case identifies actor involved in the interaction and interaction type name.

Every use-case represents interaction and system. We can use any technique to describe use-case as long as the description is short and easily understood. Use-case description helps identifying object and operation system [11].

![Use Case Diagram](image)

**Fig. 3 Use Case Diagram**

Activity diagram in the diagram that shows the flow of activities of the system. The activity diagram is seen on Fig. 4. Use-case is a technique based on scenario. It is an activity that is needed to be done in the object method. Use-case generally has a UML notation base feature to describe the system model. In the simplest form, use-case identifies actor involved in the interaction and interaction type name.

![Activity Diagram](image)

**Fig. 4 Activity Diagram**

WED also has a function as environment editor which the objects that have created in M3D.

Model consists of thousands of polygons. In M3D there are some tools at main bar can be used to start the desired model. The tools are 3 dimension structure, cubes, spheres, cylinders, and prisms. Model is composed and textured as desired by using manage skin. There are tools Scale, Move, and Rotate tool in WED. Other tools can be used to weld and emboss polygons of model and add materials.

M3D function as terrain creation is location map with continued surface. Terrain can be created by using manager tool. It can be done by importing. bmp, png, and tga image. It requires power-of-2 texture sizes. Fig. 7 shows how create terrain, by importing image from Photoshop. The bicycle model can be seen in Fig. 8.

In WED, object can be created as in Fig. 9. In addition to the bicycle, there are some inserted objects such as building.
ings and vehicles. All of created objects are called and united in WED.

To make texture in WED, inserted standard weld or imports supported weld file format is used. Texture Manager helps choosing textures that displays in dialog box at left side of WED.

Fig.11 shows the running application in which the bicycle objects is shown in the 1st perspective camera.

IV. Result and Illustration

A. System Evaluation

After the environment is implemented and united with objects and scene syntax in WED, the program is then compiled and run in one file format. Therefore the program can be run without the presence of 3D Gamestudio software.

With VR dataset, users can try bicycle path in UI environment virtually although it is not assembled with the original environment as UI proclaims.

This application has been evaluated by 10 respondents and the result has been processed using the following formula:

\[
95\% \text{ Confidence Interval} = \frac{\text{Average} \pm (1.96 \times \text{Standard Deviation})}{\text{Population}}
\]

Table 1 shows the test results.

<table>
<thead>
<tr>
<th>Question</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Perspective weld Application</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>2. Perspective weld Condition</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>3. Perspective weld Sensitivity</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>4. Perspective weld Performance</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>5. Perspective weld Accuracy</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>6. Perspective weld Relevance</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>7. Perspective weld Reliability</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>8. Perspective weld Efficiency</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>9. Perspective weld Effectiveness</td>
<td>3.5 ± 0.387</td>
</tr>
<tr>
<td>10. Perspective weld Usability</td>
<td>3.5 ± 0.387</td>
</tr>
</tbody>
</table>

In terms of the use of the 3D Wireless glasses, the following is our description of user testing. Glasses have a direct line of sight from the glasses to the transmitter. The glasses work as a filter for what we see on monitor. When the 3D is turned on, the images on the screen should appear slightly blurry or doubled to the naked eye. If we have ever seen a 3D movie in the theater this is the same concept.

The first simple way to test whether the glasses are working and the communication between the transmitter (wireless only) and the glasses is to download our program called the iD-actor (iD.exe). This can be found in the software update section of the iDimensional.com website.

To know that the 3D is turned on, first the images on the screen should appear slightly blurry to the naked eye. Lens glasses should work well. When a 3D image is being dis-
played, wears the glasses and press the ON button. In Fig. 12 shows user wears the 3D Dimensional wireless glasses.

![Image of user wearing glasses]

Fig. 11. The use of 3D Dimensional wireless glasses

From survey it concludes that environment design is good, and environment structure is good. According to respondents, VR devices such as glasses, dangle, and transmitter, are easy to use and handle and enable them to see the 3-dimension effect. The users know about VR and 3D Gemstudios well. Figure 12 shows the 3D implementation results. Respondents advise that further development and more detailed environment should be created to provide a real presence in the the VR environment.

![Image of 3D implementation]

Fig. 12. 3D Implementation

V. Conclusion

The implementation of virtual reality bicycling at University of Indonesia green eco-campus environment using 3D Gemstudio software has been shown in this paper. The user test results show that users are relatively satisfied with the 3D built environment. However, further more detailed implementation is necessary by adding more features. The use of 3D Glasses for the VR experience has provided an extra experience using different devices to create a more real environment.

Future work will involve improving the virtual reality environment of the bicycle track and all buildings at the University of Indonesia’s Depok campus. We will also explore other VR application development platforms such as Unity 3D. We plan to put the system in public spaces in which students and other users can use the VR system. This is also will promote the presence of environment by using the free bicycling facility at the University of Indonesia.

References