DEVELOPMENT PROCESS OF A DRIVING SIMULATOR

Mohd Khalid Mokhtar 1, Mohd Shahrizal Sunar 2

1 Department of Computer Graphics, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia
81310 Skudai, Johor
khalmokh851027@yahoo.com.my 1, shah@fsksm.utm.my 2

ABSTRACT

Advancement in Computer Graphics and visualization technology brought real world situation into simulated environment. Simulated environment allows us to drive in controlled and repeatable environment. This paper presents studies on creating and processing simulation environment for driving simulator. Simulation environment involves several developments processes to become a complete set of driving simulator. The development processes are divided into creating the software and integrate the software with the hardware. The main contribution of developing this driving simulator is for development and research programs in Computer Graphics area. The result of development driving simulator encourages students to apply anything that they learned by producing a product that can be used by user. This paper describes a driving simulator belongs to Universiti Teknologi Malaysia (UTM). Reconstruction of the main area of UTM became the first virtual environment for this driving simulator. Equipment, system architecture and interfacing element of the simulator are presented.

Keywords: Computer Graphics, Driving Simulation.

1 INTRODUCTION

A driving simulator is a device that allows user to feel like real experience of driving an actual vehicle within controlled and repeatable environment. Driving simulator is used in many researches such as studying the interaction of a driver and vehicle and for developing new vehicle systems, human factor study, and vehicle safety research by enabling the reproduction of the actual driving environments in a safe and tightly controlled environment [1]. A complete set driving simulator consists of physical mockups as examples steering wheel, gearshift and pedals. In an immersive driving simulator example in Figure 1, it is most important to provide a realistic feeling in a real moving vehicle. The use of simulator for driver training is much less expensive compared to pilot simulator because the total student involve in car driving training is larger than other vehicles simulator.

UTM driving simulator is a project used to improve Computer Graphics study by learning by doing. This project allows student to apply and improve their knowledge of Computer Graphics. Many researches can be done in this project for example 3D scene management, collision detection, crowd rendering, global illumination and artificial intelligent (AI). In this paper, we first reviewed related research on driving simulator. Then in section 3 we elaborate the challenges in developing the driving simulator. Next, section 4 describes the hardware architecture of driving simulator and we continue with system design in section 5. Section 6 describes about rendering process in driving simulator virtual environment.

Figure 1. UTM driving simulator.

2 RELATED WORKS

Driving simulator is not a latest research, especially its development and application. As early of 1900s, driving simulators start having their roots on flight simulators. It has begun to appear in primitive forms in the 1970s. The advent of computer technologies brought Daimler-Benz to launch a high-fidelity driving simulator in 1980s,
since then brought many automotive makers and research institutions worldwide to develop their own simulators [2]. Soon, computer generated imagery was more extensively being introduced in driving simulation research on the early of 1980s. Computer-generated image systems advancement went through several stages in the 1980s, from very angular and lack of shading and detail to the appearance of photo-driven texturing. Through a significant software development cost realistic images look like real [3]. Driving simulator nowadays also used for virtual tour which people can share experience and knowledge about a city [4]

3 CHALLENGES IN DEVELOPING DRIVING SIMULATOR

Development of driving simulator involves two parts: a) Software b) Hardware. Development of software includes several processes such as 3D modeling, VC++ programming, OpenGL programming and socket programming. The hardware divisions can be exemplified by multi-screen for 3 channel LCD TV, sound, steering, gear and networking. Skills are needed in both parts; especially for computer graphic study requires expertise on VC++ programming, OpenGL programming and little knowledge on 3D modeling. For hardware part, skills are needed in mechanical, electrical and basic in networking.

In developing a driving simulator, four basic components are required [5]:
- Simulation of the physics of the vehicle and its interaction with road surface.
- Simulation of the surrounding environment, including other vehicles.
- Integration of informative systems and displays that enable subjects to interpret the state of the vehicle model, e.g. sound rendering, instrument panels, and motion base.
- Integration of control devices, e.g. steering wheel, accelerator pedal, brake pedal, shift lever, clutch pedal, and other vehicle controls.

4 HARDWARE DESIGN FOR DRIVING SIMULATOR

The development of driving simulator in UTM air aiming for the implementation of any research within the Computer Graphic area to a product and produce a good driving training lesson to user. The hardware architecture consists of three clients and a server. All clients are connected to a widescreen LCD color TV with a size 37 inch. Each of the clients equipped with a high-end graphic card to produce high quality display of virtual environment. The clients and server running the same simulation but the server is used to send synchronize messages to each client. The differences between each client are camera position and look at vector in rendering processes. Connection between clients and server is through Ethernet LAN. This hardware architecture almost resemble system architecture mid-range driving simulator developed at the University of Skovde [6]. Figure 2 shows the overall hardware architecture of the driving simulator.

![Figure 2. The hardware architecture design of driving simulator](image)

4.1 The Driver Atmosphere

The atmosphere of the driver is half-cut car of Proton Waja. Figure 3 shows the driver atmosphere.

![Figure 3. The Driver Atmosphere.](image)

The use of half-cut car allows user to feel like in real driving situation. It will enhance realisms for the driver by using actual devices. The half-cut car only took the driver portion of a real car.
4.2 Screen Projection

Wide screen LCD color TV with a size 37 inch width provides 1360x768 high resolution images to each TV. In figure 4 shows how to arrange the position of screen to driver.

Field of view (FOD) with high resolution graphics in the projected is important for the driver to have a realistic driving feel and react to the driving environment precisely [7].

5 SOFTWARE DESIGN FOR DRIVING SIMULATOR

The architecture of software has been developed to support scalability, extendibility and flexibility. The capability of software to be enhanced means that in the future more upgrading can be done. Programming language used in this software is C and graphic subsystem chosen is OpenGL. The use of OpenGL as graphic API is because it supports variety of platforms (Konyha 2000). This software design is almost the same as TODS system design [7]. This system design can be divided into 3 looping process: 1) Rendering Loop 2) Operation Loop 3) Physic Loop. Figure 5 below shows driving simulation system design is a human-in-the-loop mode.

Operation Loop, driver of driving simulator sends input control (steering wheel, break, and throttle) operations to the inner loops. Operation Loop depends on response from user through simulator control. Then in Rendering Loop, GPU renders the virtual scene via graphic pipeline. At the same time GPU apply effect to scene using Vertex Shader and Pixel Shader. Lastly in Physics Loop, the Physics engine used to calculate the driving car’s position, velocity, acceleration, and orientation.

6 RENDERING

The virtual driving simulator environment consists of static universe, dynamic objects and interior of driver’s vehicle. The static universe can be building, trees, road and others. The dynamics objects can include any moving objects in virtual scene like cars, people, and crowd. Complex virtual scene will contain many thousands of polygons, which need more graphic processing power and more computation cost to render the scene. Even on latest graphic hardware processing, the increase in complexity in virtual environment will increase computational power. The complex scene is needed to manage so it can be efficiently render in real time and avoid memory leak. Implementation of 3D scene management techniques may help to reduce computational burden in complex driving simulator environment. In this project, Open Dynamics Engine (ODE) physic engine was used in helping in collision detection, suspension, interact with environment and artificial intelligent (AI). Usage of graphic processing unit (GPU) rendering assists in producing special effect to achieve realisms.

6.1 3D Scene Management

3D scene management can be defined as algorithms and methods that select only the polygons that are needed for viewer depending on location and orientation of virtual camera [8]. Developed scene management for the driving simulator applied: (a) Space partitioning (octree), (b) image-based and (c) culling. Octree is pre-process technique that create large bounding box and subdivide it until meet some condition. After 3D model loaded by 3D loader, octree will use the information from 3D loader to create octree for the loaded 3D models. Visible node from octree tested using frustum culling and if a node not inside the frustum, the node will not be rendered. Figure 6
shows us octree that used in virtual environment for driving simulator.

![Octree](image1.png)

**Figure 6. Octree.**

Image-based used in the virtual system allows less computation cost by cut the usage of polygon to be rendered by GPU. The usage of image-based rendering for trees in driving simulation is seen in Figure 7.

![Image-based trees](image2.png)

**Figure 7. Image-based trees.**

Culling technique used is far plane culling. Far plane culling is technique that removes object that completely behind the plane [10]. Far plane culling has minor disadvantage when objects in distance. Solution for this problem is the use of fog to hide behind the far plane.

There are rooms for improvements to improve the 3D scene management used by the virtual system. More complex virtual scene more efficient techniques will be required. As example, the use of GPU in helping to reduce computation cost to render complex virtual scene. The usage of 3D scene management allows complex virtual scene to be interactively rendered in real-time.

### 6.2 Physic and AI

ODE is an open source physic engine that has contributed in many applications such as car simulator, flight simulator and games. ODE is easy to use and fast to develop a virtual system. Main function of ODE is used to simulate articulated rigid body dynamics for virtual systems. For driving simulator, rigid body means the body of car. All part of a car connected using joints. A physic engine required to stimulate physics of real car to the car in virtual system. The usage of variable such as mass and velocity in physic will approximate what happen in real life to a virtual environment. Simulation process commonly using following steps [9]:

1. Create a dynamics world.
2. Create bodies in the dynamics world.
3. Set the state (position etc) of all bodies.
4. Create joints in the dynamics world.
5. Attach the joints to the bodies.
6. Set the parameters of all joints.
7. Create collision world and collision geometry objects, as necessary.
8. Create a joint group to hold the contact joints.
9. Loop:
   (a) Apply forces to the bodies as necessary.
   (b) Adjust the joint parameters as necessary.
   (c) Call collision detection.
   (d) Create a contact joint for every collision point, and put it in the contact joint group.
   (e) Take a simulation step.
   (f) Remove all joints in the contact joint group.
10. Destroy the dynamics and collision worlds.

Fully utilize the capabilities of this physic engine help to improve realism of a driving simulator. This is importance to make a driving simulator to act like a real driving such as speed and collision of a car.

### 6.3 GPU Assistance

OpenGL provides its own shader language called GLSL. GLSL manipulates vertex shader and pixel shader to produce authentic effect. UTM driving simulator use GPU rendering in helping to produce special effect such as shadows, bloom effects and rear view mirrors.

One part of GPU rendering still not touched in this project is the usage of GPU assistance to reduce burden on CPU. More can be done to use another GPU effect or GPU assistance in this project to utilize capability of GPU rendering. Figure 8 shows the usage of GPU assistance to make special effects.
7 CONCLUSION AND DISCUSSION

This paper will lead to the development and researches in Computer Graphics to the complex virtual environment of a driving simulator. The potentials of improvement for driving simulator will lead used to another real time graphic system. The developments and researches are important to develop a system that able to achieve fidelity and realism like a real world. This project more focus on Computer Graphics area, where more researches and improvements to this driving simulator soon, especially in rendering processes such as 3D scene management, physic / AI and GPU assistance. Produce a quality software will be able it marketed to used in many area such as car companies and driving training agencies.

REFERENCES

[This page intentionally left blank]