CPT205 Assessment 2 Report

3D Modelling Project

Mingyuan Li(2145618)

ICS

December 16, 2023

1 Introduction

1.1 Task Overview

The assignment is to design a 3D sense with a number of objects. The scene should have both animated and static elements, and actions can be triggered by keyboard or mouse inputs. By using OpenGL only with the freeglut library, achieving a high-quality visual effect using techniques like 3D geometry creation, viewing transformation, constructive solid geometry (CSG), hierarchical modeling, lightening and materials effects, texture mapping, animation and so on.

1.2 Project Design Overview

This 3D scenario is centered on the CB of XJTLU, which is inspired by assessment 1. The exterior wall of CB is composed of a multi-layer partition structure, and the surrounding scene has roads and rivers [1], as shown in Figure 1. Therefore, its rich scene elements make it possible to build a scene suitable for task requirements. Specifically, a ground surface is used to define the entire scope of the scene, with multiple lanes of traffic and moving cars on top of the ground, a river set in the ground, multiple rows of trees, and the main element: the CB. The entire scene can be manipulated using the mouse or keyboard to control the direction and point of observation and to maneuver the moving car. The scene uses a variety of different textures and rich use of materials and lighting. Three different light sources were set up to achieve a pink sunset color like Figure 2.

Figure 1: The CB in the real world

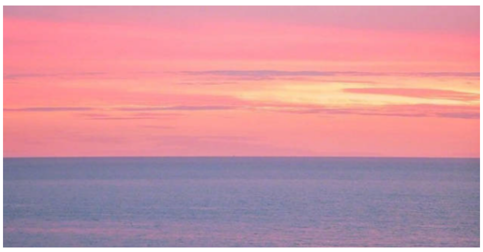

Figure 2: Pink sunset scene in the real world

2 Software Environment Configuration

The software environment closely adheres to the OpenGL and MS VC++ environments as specified by lab 1. Only the freeglut library and the most basic C++ libraries are used throughout the entire project.

3 Design and Features

This section is for illustrating the design purpose and features of each elements. The following details the method and reasoning behind the design.

3.1 The CB

The entire CB is constructed in 3 parts: the square theme, the irregular polygonal base, and the exterior partition body. The main body is a cube with four glass-like textures around its surface. The base is made up of a number of trapezoids of different sizes, which simulates the appearance of the low part of CB. The partition body consists of a number of thin rectangles of different sizes, whose position corresponds to the real alignment. To emphasize the texture of the main cube, the base and partition are not textured and are used grey material to represent it's real color.

3.2 The Trees

A tree consists of a trunk represented by a cuboid and leaves composed of two sets of cylinders. The leaves are formed by two sets of cylinders, each of which consists of four cylinders distributed symmetrically around a central point. Then, two sets of cylinders are stacked to simulate the effect of leaves stacked and interleaved. Both parts have the material colors they should have.

3.3 The Cars

There are two kinds of cars in the scene: regular cars and Tesla cybertruck. Both cars consist of four tires, a main body (chassis), and a protruding cockpit. The difference between the two cars is the appearance of the cockpit. The tire is made of a cylinder and the body of the car is made of cuboids. The cockpit of the normal car is a trapezoidal body, while the cockpit of the cybertruck consists of a triangle shape from front to end to simulate the real construction.

3.4 The Roads

There are three roads in the scene, including two motor roads and one campus sidewalk. Each is made of thin cuboids with a texture on the surface that represents the appearance of the road.

3.5 River

To simulate a river to the west of CB, use blending for transparency and alpha testing for fragment discarding to draw a river below the ground. The river is made of cuboids and has a corresponding material to represent the flow of water.

4 Implementation and Graphic Techniques Details

This section will illustrate the detailed implementation and corresponding techniques [2].

4.1 Base method

To save space and make the code more readable, basic helper functions are defined to be called repeatedly to draw the corresponding 3D objects part of the cuboid, cube, and cylinder, such as `renderTexturedQuad`. Then the helper functions are used to assemble the closed polygon, and several different base polygon functions are generated, which can accept different scaling ratios and different texture mapping rules.

4.2 Viewing and Projection

The project uses perspective projection to simulate the effect of human eyes seeing the real scene. In addition, the interaction of manipulating the position and direction of the observation point is added, so that they can be moved to different directions and positions to observe the scene to meet different detection requirements.

4.3 Transformation

Various geometric transformations are used in the project, mainly scaling, translation, and rotation. The base objects created in the project, such as rectangles and cubes, are all of the same orientation and size, so the shape of the final displayed elements is achieved by translation and scaling, leveraging with `glPushMatrix()` and `glPopMatrix()`.

4.4 CSG

The CB in the project used CSG to simulate the effect of the real appearance of the so-called Taihu Lake stone partition structure. The main strategy is union, which involves 3 parts, a base part, a main body, and a partition part. By using CSG, they can be stacked or overlapped into the desired shape, as seen in Figure 3.

4.5 Hierarchical Modeling

To organize the objects in the scene into a hierarchical structure, making the relationship between the objects clearer and easier to manage and control, hierarchical modeling is used extensively in the project. The most

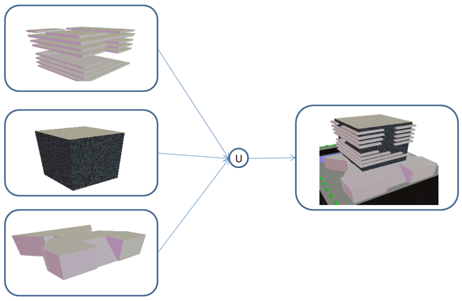

Figure 3: The CSG structure of the CB

representative is the vehicle in the scene, where the whole vehicle serves as the root node, and the chassis is a subnode connected to 4 sub-nodes, which are 4 wheels in different positions. And it's structure shown in the Direct Acyclic Graph (DAG) below.

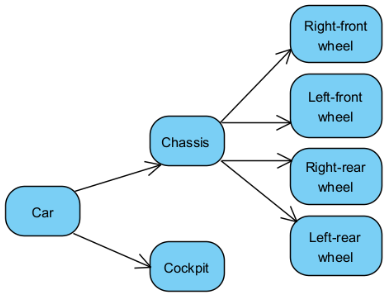

Figure 4: The SAG structure of the car

4.6 Lightening and Materials Effects

Elements in the project use `glMaterialfv` to set different material parameters such as `GL_AMBIENT`, `GL_DIFFUSE`, `GL_SPECULAR`, and `GL_SHININESS`. Meanwhile, `glLightfv` is used to set the ambient light, specular reflected light, and diffuse light in the scene to affect the display of object materials. Finally, three different light sources were set up in to achieve a pink sunset color.

4.7 Texture Mapping

There are 4 different textures used in the project, all of them conforming to the properties of the corresponding objects [3] [4]. Imported textures are bound to a fixed surface using `glBindTexture` and `glTexCoord2f` when creating the base object.

4.8 Depth Testing and Blending

For the rendering of the river scene, the project used depth testing, blending, color masking, depth masking, and alpha testing to achieve a more realistic visual effect. Specifically, enabling depth testing ensures that objects are properly occluded according to their depth in the scene. Meanwhile, `glEnable(GL_BLEND)` and `glBlendFunc()` are used to achieve a reasonable blending of transparent objects (water layer) and background. Also, the Alpha test is enabled so that the water layer can be cropped based on its alpha value, giving it a transparent effect.

4.9 Animation

To add movement animation to the cars in the scene, a parameter updated by a timer is added to each car's `glTranslatef` to achieve the movement effect. After moving to the end of the scene, initialize the movement parameter to achieve the effect of repeated vehicle passes. In addition, on the other road, to simulate the effect of a traffic light, the car will no longer move when moving to the intersection of the road.

4.10 Interactions

The project uses a number of interactive features, including mouse clicks, drags, scrolling, and keyboard input. Enabled 4 keyboard or mouse-related methods for interactive effects. The detailed interactive keys are as follows:

- **Camera interaction**
  Press 'a' to move the camera position to the left.

  Press 'd' to move the camera position to the right.

  Press 'w' to move the camera position to the up.

  Press 's' to move the camera position to the down.

  Press 'i' to change the camera orientation up.

  Press 'k' to change the camera orientation down.

  Press 'j' to change the camera orientation left.

  Press 'l' to change the camera orientation right.

  Press 'z' to move the camera position closer.

  Press 'x' to move the camera position farther.

  Press 'o' to zoom in the scene.

  Press 'p' to zoom out the scene.

- **Exit**
  Press '0' to exit the program.

- **Pause and vontinue vehicles movement**
  Press 'q' to control all vehicles to pause or continue movement.

- **Mouse interaction about the camera**
  Pressing the left mouse button and dragging the mouse also change the camera orientation.

  Sliding the mouse wheel to zoom in and out the scene as well.

5 Program Execution and Results

As described in the previous section, there are several interactions involved in the program, so it is necessary to give specific examples of the program execution. The following are multiple samples of the scene observed from different viewpoints, which contain the situation of pausing the movement of the cars.

Figure 5 shows the view at initialization, and all objects have been viewed except for the vehicle behind the CB, which is obscured.

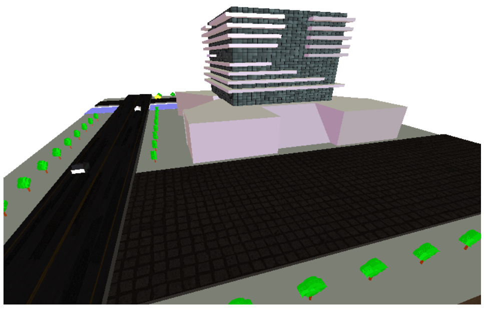

Figure 5: The view at initialization

Figure 6 shows the scene of an observation point shifted from the viewpoint at initialization to the right. This operation is accomplished by entering the 'd' on the keyboard. The scene can now be viewed to the north and east of the CB, and it can also be noticed that the position of the car has changed.

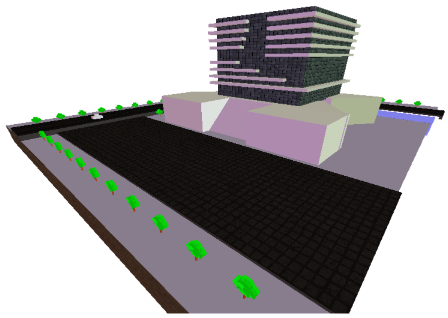

Figure 6: The view at the right of initialization

Figure 7 shows a viewpoint moving in the direction of the initial point forward, accomplished by pressing 'x' on the keyboard. The viewpoint orientation was also adjusted by sliding the mouse. It can also be noticed that the vehicle is in the same position as in Figure 5, which is accomplished by typing the 'q' on the keyboard.

6 Conclusion

By using multiple OpenGL methods, the project successfully produced a 3D scene about the CB and surroundings in the XJTLU and met the task requirements. The

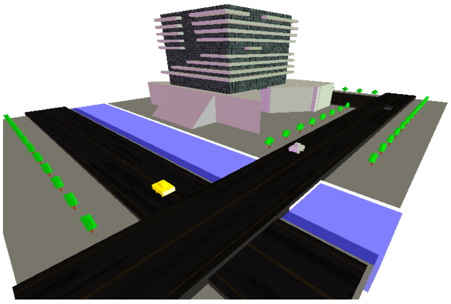

Figure 7: The view at the forward of initialization

project can achieve those goals through the 3D geometry creation, viewing transformation, CSG, hierarchical modeling, lightning and materials effects, texture mapping, animation, and so on. The project is defined by a ground surface that includes several traffic lanes and moving automobiles on top of it, a river carved out of the ground, numerous rows of trees, and the central component—the CB. The direction, position of observation, and movement of the moving automobile can be controlled using the mouse or keyboard. The scene makes extensive use of materials, lighting, and a range of textures. Finally, by setting the light source, a complete scene of a corner of XJTLU in a pink sunset environment is built.

Acknowledgement

The project's read image function(`ReadImage`), texture setup function(`setTexture`) and the raw arrays of vertices come from the Github. In addition, the basic quadrilaterals generation was inspired by it.

References

[1] (Access: Dec. 16 2023) Xi'an jiaotong-liverpool university - our campus. [Online]. Available: https://www.xjtlu.edu.cn/zh/study/why-study-at-xjtlu/our-campus

[2] (Access: Dec. 16 2023) Opengl functions. [Online]. Available: https://learn.microsoft.com/zh-cn/windows/win32/opengl/gl-functions

[3] (Access: Dec. 16 2023) Ambientcg. [Online]. Available: http://ambientcg.com

[4] (Access: Dec. 16 2023) Poly haven. [Online]. Available: http://polyhaven.com