



Xi'an Jiaotong-Liverpool University
西交利物浦大学

Assessment 2 – 3D Modelling Project	
Module Name	Computer Graphics
Module Code	CPT205
Student Name	Yiming.Liu
Student ID	1930131
Degree Programme	Information and Computing Science BSc

1. Introduction

Computer Graph (CG) technique had been popular in people's daily lives, such as videos, films producing, the object recognition, and so on. To support the basic theories of CG concepts, that request a tool can call the relative methods, such as OpenGL.

This assignment is in order to design a 3D scene using the OpenGL libraries. And to meet the requirements that the object will get appropriate texture, adding real-looking lights, and the moving actions. That request us to familiar with the C++ language, OpenGL API, and the operations of the Visual Studio. To get such an ideal result, the author had studied relative linear algebra knowledge, and exercised writing C++ codes. With these conditions and learned from the online resources, in the end, with the following methods, the author successfully build a nice-looking 3D scene, a mini-planet.

2. Methodology

2.1. Lighting effects

To simulate the real light such as sunshine (parallel light), diffuse reflection, and point light source, the code should design before well organizing the 3D coordinates and related attributes of the lights.

2.2. Moving actions

Because that the matrix calculations are slow, it is better to reduce the matrix calculations. Therefore, a common strategy is to moving the viewing point to simulate the real moving action rather than moving the model that the code designed.

Also, some special effects like zoom in and zoom out, the code can simply add a more scale attribute for each part of the scene to adjust the size of the final model.

2.3. Texture effects

Two ways to add the texture effects. First, use the RGB to design the texture. Second, load the picture such as PNG documents to decorate the model.

In order to design a nice-looking scene, the author had self-learned related project from the internet, and self-writing a code library to load OBJ model documents with the MTL material (texture) documents.

For designing the model, the author using the professional 3D design software named Blender. To shorten the produce period, the author also downloaded some open source resources, some basic OBJ models such as person, football, tree, and so on. The author built a local blender project and paint the model and export it for a new OBJ with MTL. Then load it in to the memory of the C++ program and call OpenGL methods to draw the model.

3. Implementation

3.1. Object-Oriented Programming (OOP)

According to the requirements that the assignment can only use the basic C++ STL and the OpenGL static library, the author use the C++ STL to self-designing a simple OBJ loading tool.

This needs some OOP knowledge, the author designed a self-writing library, which includes the picture loading methods, the material document loading methods, and object document loading methods.

3.2. Functions of the Display Window

OpenGL has the related light, texture, and display functions. The author designed the basic frame under the **main.cpp**. The author designed a mini-planet with clouds, vehicles (plane, car), person, football, trees, mountains, and volcano.

4. Readme

There is also a **README.md** document in the download directory.

4.1. Initial Configuration

Move **all the files and folders** into the existed Visual Studio directory.

Add the **main.cpp** into Visual Studio project then compile and run the main.cpp.

4.2. Mouse Interactions

Keeping press the left mouse button and left or right move. Then the model will be left or right rotate (rotate in the x plane).

Keeping press the left mouse button and front and back. Then the model will be up or down rotate (rotate in the y plane).

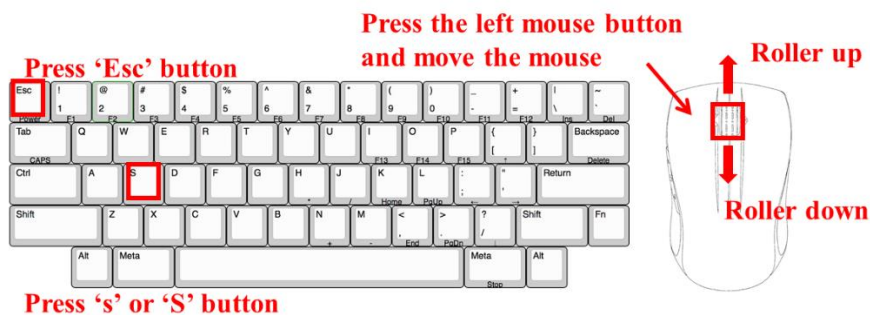
The two operations above can be executed at the same time. For example, the tester can click the left mouse and drag the mouse moving for free to moving the model.

Mouse roller can control the size of the model displayed. Roller up to zoom in and roller down to zoom out.

4.3. Keyboard interactions

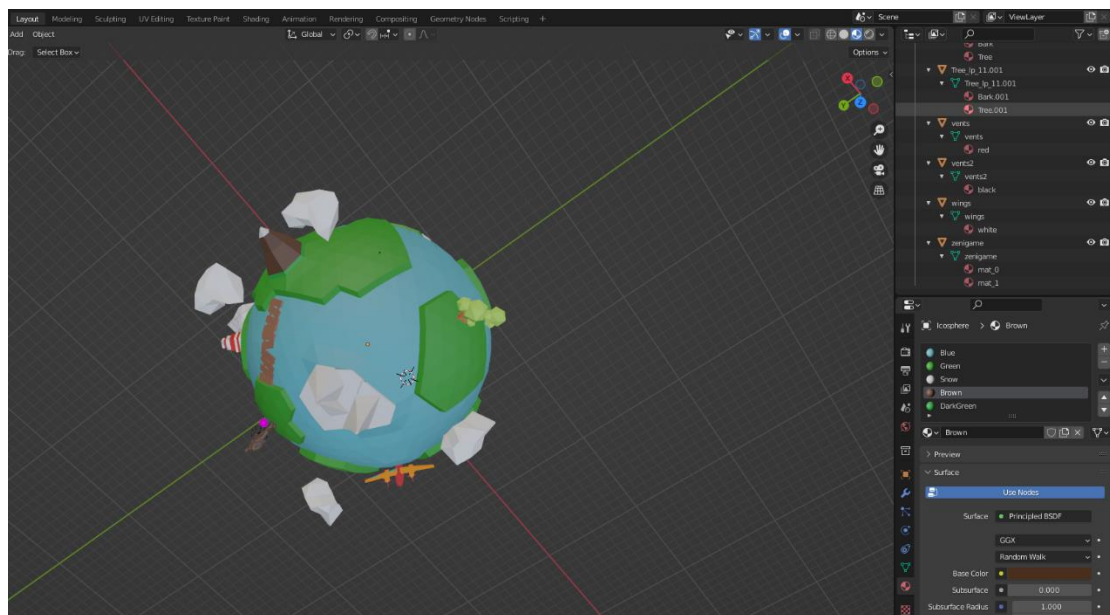
Press the 'Esc' button to quit the display window.

Press the 's' or 'S' to shift the model's texture. Press once to display the texture, double press to remove the texture.

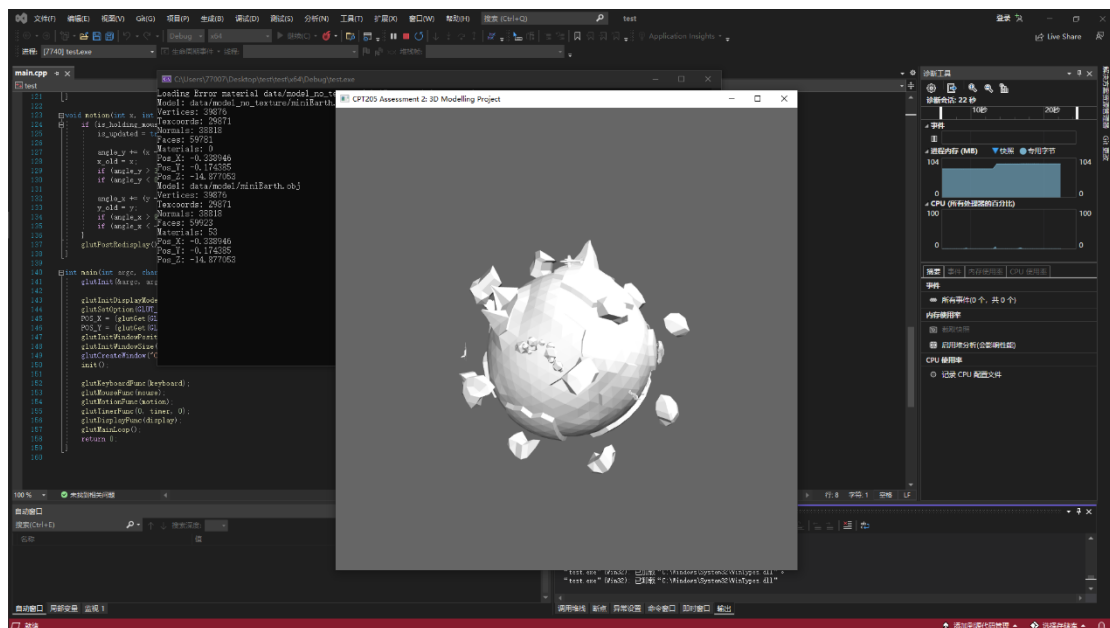


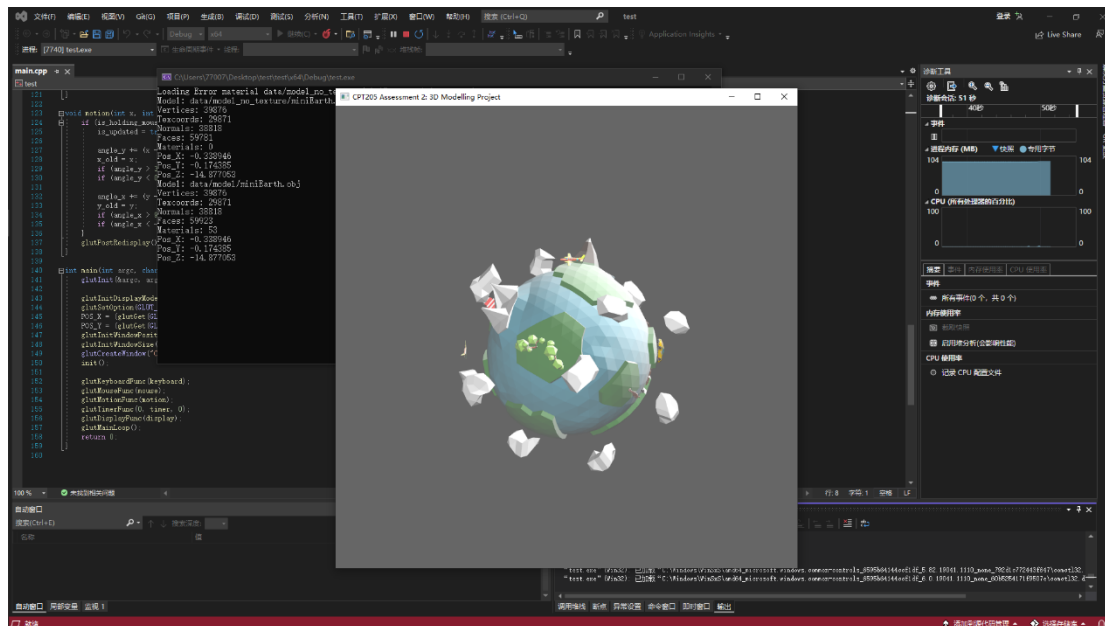
5. Testing and Result

5.1. Screenshots (Blender model building)



5.2. Screenshots (C++ via OpenGL display outcome)





6. Conclusion

One expected function not achieve is model moving via mouse combination operation. Due to the time limitation, the project had not achieved a more difficult interactions clicking with dragging operation on mouse. This mouse combination operation supposed to designed to change the absolute coordinates of the model. For example, some other advanced 3D design software such as Blender, Maya, Unity3D, and 3D Max, can simply moving the models via the mouse.

Another drawback is that the light effect is not pleasing to the eye enough, the diffuse light is too bright that the original color of the model had not displayed 100%. It could be better, if there are more time to adjust the attributes of the lights.

Reference

- [1] <https://blenderartists.org/c/support/materials-and-textures/38>
- [2] <https://github.com/WHKnightZ/OpenGL-Load-Model>
- [3] <https://www.runoob.com/cplusplus/cpp-tutorial.html>