FINAL REPORT FOR COMPUTER GRAPHICS 100433

TEAM MEMBERS

1651047	计算机 1 班	杨文翔
1652248	计算机 2 班	秦睿
1652265	计算机 2 班	左心笛
1652635	电子信息工程	贾育文
1652289	计算机 2 班	杨宇航
1651588	计算机 2 班	高帅军

PROJECT TITLE

An Interactive Virtual Reality Tour of CAUP (College of Architecture and Urban Planning).

ABSTRACT

With the popularity of immersive VR games, the sense of surprise that VR technology brings to us has gradually reduced. Thus, interactive virtual reality will be the next step in immersive VR. As developers, inspired by the current accomplishment of 3D modeling, human-interaction techniques and VR game experiences, our team decide to create a system which allows people to tour around the campus by airplane.

This project supports that through one or two minutes of automatic driving, or interactive control, the school building model can be observed through the perspective of the aircraft model. Together with other functions, like weather effects, the audience experience is optimized.

In order to achieve requirements above, the following technologies are used: 3D transformations, model loading, particle system, shading and lighting. There are two related libraries: the OpenGL Extension Wrangler Library (GLEW) and Simple and Fast Multimedia Library(SFML).

MOTIVATION

Interactive virtual reality is the next step in immersive VR. It eliminates passive viewing experiences and increases engagement, allowing people to see what is happening around them.

As more and more people experience this type of virtual reality, and as the initial "wow factor" of VR diminishes, it will be important for developers to create interactive virtual reality experiences and virtual reality creators will need to add ways to consume multiple forms of media and choice.

Inspired by the current accomplishment of 3D modeling, human-interaction techniques and many real game experiences, our team figure it would be breath-taking to create a system which allows people to tour around the campus without the need of being there.

Combing the preselected routes and interactive control, users can have more friendly experiences touring around. The idea of weather adjustability exemplified by several particular weather, namely raining, snowing and so on, will surely improve the enjoyment of the journey. Hopefully, if the project goes as expected, it may even show its commercial potential.

Goal of the Project

- Provide an immersive VR experience to players exploring Tongji campus by flying a drone.
- Different way of exploration, one is automatic exploration following pre-selected routes, the other is interactive flying controlled by the player.
- Different sights as the aircraft travels, generating aircraft perspective.
- Randomly generating particular weather, corresponding to the light, shadow and sound changing.
- A refined and apt aircraft model, compliance with basic laws of physics.
- Focus on the particular one of three aspects—weather, lighting and shading

Scope of the Project

Included functions

Tongji Model Establishment

The model will be built in two ways:

1. First the model should be constructed in SketchUp pro, which is a modeling software & can output the model into OBJ format. The model should include the building, streets & trees. Its model design is based on the real Tongji University.
2. Second we need to load the OBJ file into memory and then transfer the data into Opengl buffer object.(Here we made a mistake, which we will talk about later on).

The school should be 4000*5000*4000, and we set up a boundary so that we would not move the aircraft out of the space.

Each building should have an estimated parameter according to their real model in our school.

The aircraft is quite small and we set the initial parameter to 2*2*3 so that it can enter the building and see the more detailed inner spaceDrone modeling

Weather System

We tried to make a more complex weather system for our work. We may have several choices like sunny, foggy & dark.

We can also set up the color temperature to be more comfortable.

Aircraft Modeling System

Another big part of the system is about the aircraft modeling system. This model have a lot of features:

1. Position: the position of the aircraft
2. Direction: the direction the plane moves forward
3. Speed: the plane should keep in low speed so the whole scene details can be ovserved.
4. Rotation Matrix: to accomplish the rotation of the airplane

The Aircraft is the only moveable model of our system. We try to make it very uniquely designed.

It can move forward, backward, upward and downward & rotate.
It can also enter the building.

The camera also follows the aircraft and is able to zoom in & out with Ctrl keyboard.

Restriction

Size of the aircraft

The plane could not be too large to enter the building and the room in the building.

Map Boundary

A boundary should be built for the map and the building so that the aircraft will not pass through the model or go out of the map.

Involved CG Techniques

3D Transformations

Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation using matrix operations. In our project, 3D transformation is used in the rotation of the airplane and translation of the terrain.

Model Loading

The model of the buildings of the University is created using SketchUp, a 3D modeling tools that allows the creation of complicated shapes and apply textures to them.

A very popular model importing library out there is called Assimp that stands for Open Asset Import Library. Assimp is able to import dozens of different model file formats (and export to some as well) by loading all the model's data into Assimp's generalized data structures.

Texture Mapping

Texture mapping is a method for defining high frequency detail, surface texture, or color information on a computer-generated graphic or 3D model. To create a skybox, we only used texture mapping in its original sense that simply wrapped and mapped pixels from a texture to a 3D surface.

Shader

A shader program is a small program to control the graphics pipeline, in particular:

- Vertex shader controls vertex transformation
- Fragment shader controls fragment shading

In our project, the shader program that consists of a vertex shader and a fragment shader is written in GLSL. We use it to implement:

Lighting

Lighting in OpenGL is based on approximations of reality using simplified models that are much easier to process and look relatively similar.

Phong reflection model

Phong reflection is an empirical model of local illumination. It describes the way a surface reflects light as a combination of the diffuse reflection of rough surfaces with the specular reflection of shiny surfaces. It is based on Phong's informal observation that shiny surfaces have small intense specular highlights, while dull surfaces have large highlights that fall off more gradually. The model also includes an *ambient*term to account for the small amount of light that is scattered about the entire scene.

Libraries

The OpenGL Extension Wrangler Library (GLEW)

The OpenGL Extension Wrangler Library (GLEW) is a cross-platform open-source C/C++ extension loading library. GLEW provides efficient run-time mechanisms for determining which OpenGL extensions are supported on the target platform. OpenGL core and extension functionality is exposed in a single header file.

Simple and Fast Multimedia Library(SFML)

SFML is a cross-platform software development library designed to provide a simple application programming interface (API) to various multimedia components in computers, including:

Project Contents

Camera Module

Perspective conversion

Adjustable the angle and position of the camera to explore different building shapes and aircraft models from a wide range of perspectives.

Interaction
Players have control of the drone.

Architecture Modeling
Due to the hardware system drawbacks, the architecture model loaded into the program became smaller.

Shader
Using Shader to imitate the appropriate levels of material texture and color within the architecture model.

Lighting
Using Lighting in OpenGL to imitate the actual light and shadow effect of the architecture.

Model Loading
Combining the model building using the sophisticated SketchUp and the bridging OpenGL library Assimp.

Aircraft Model

Route design
Design the flying route to travel around the architecture and avoid obstacles

Size design
Design the suitable size to fit the real flying route and actual architecture shape and indoor structure.

Weather System

Fog Weather
Try to imitate the fog weather use perspective projection and lighting system.

Implementation

Week 3-5
Learn to use GLSL, with which we can programe shaders that are executed on the GPU. Establish interior model of Tongji University building by using SketchUp. Import SketchUp to OpenGL by Open Asset Import Library (Assimp), which is a portable Open Source library to import various well-known 3D model formats in a uniform manner.

Week 6-8
Build a flight simulator with GLEW which helps in querying and loading OpenGL extensions, and SFML which provides a simple interface to the various components of PC, to ease the development of games and multimedia applications.

Built airplane, camera, model loader, mouse controller and keyboard controller modules.

Week 9-11
Combine the flight and indoor models.

Build shadow, shader and terrain renderer module.

Week 12-13
Add other functions like weather, automatic flying, etc.

Fog weather imitation and interesting perspective chaning.

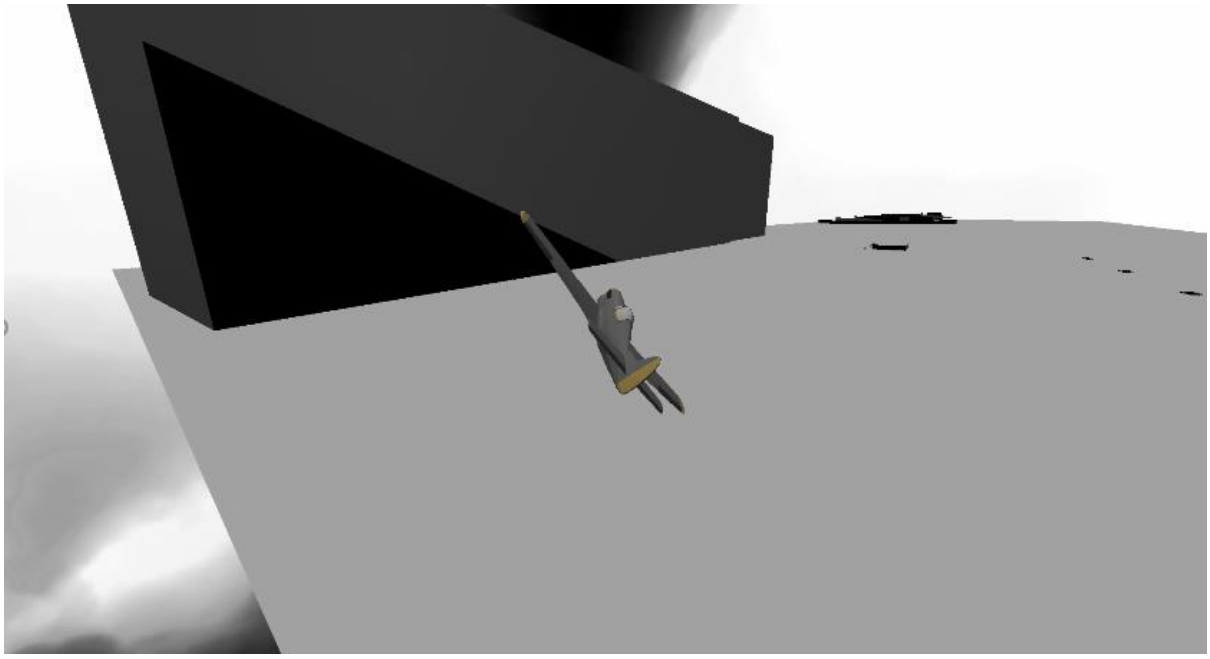
Week 14
Continuously improvement and fine tuning of the system.

Results & Later Improvement

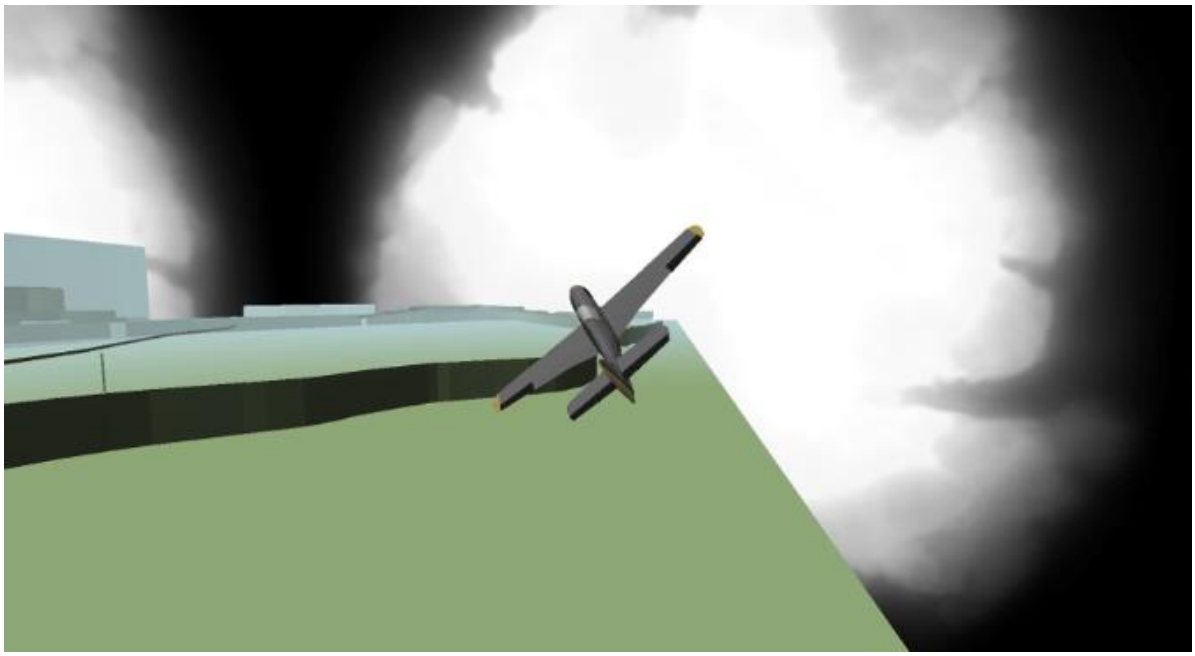
Results

Aircraft Model

Pre-shading

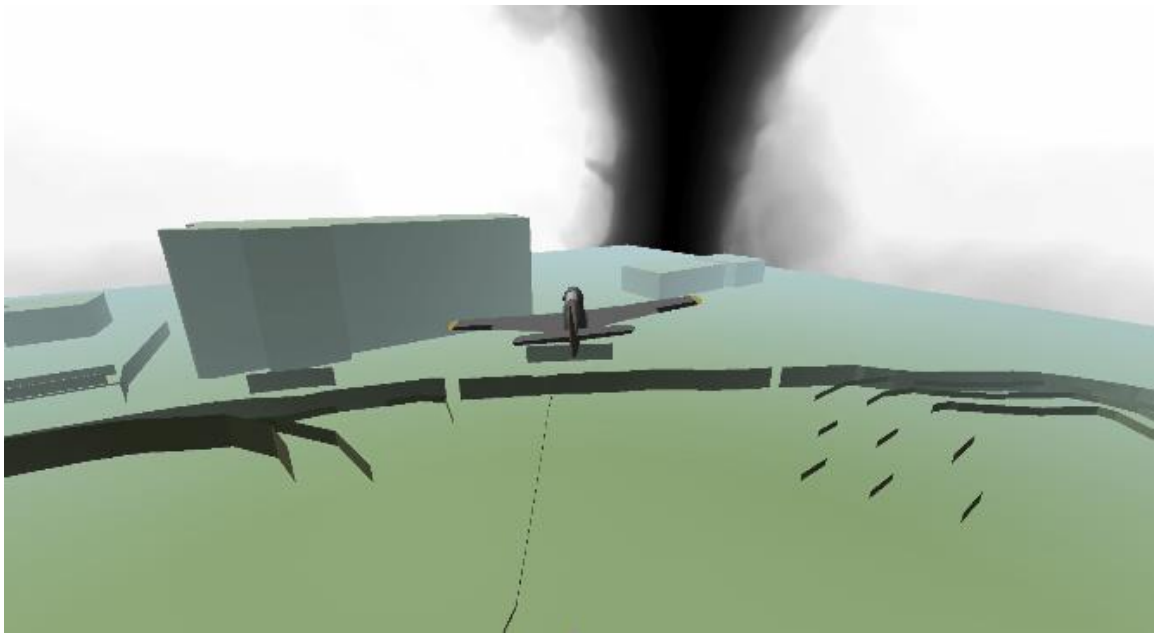


After-Shading

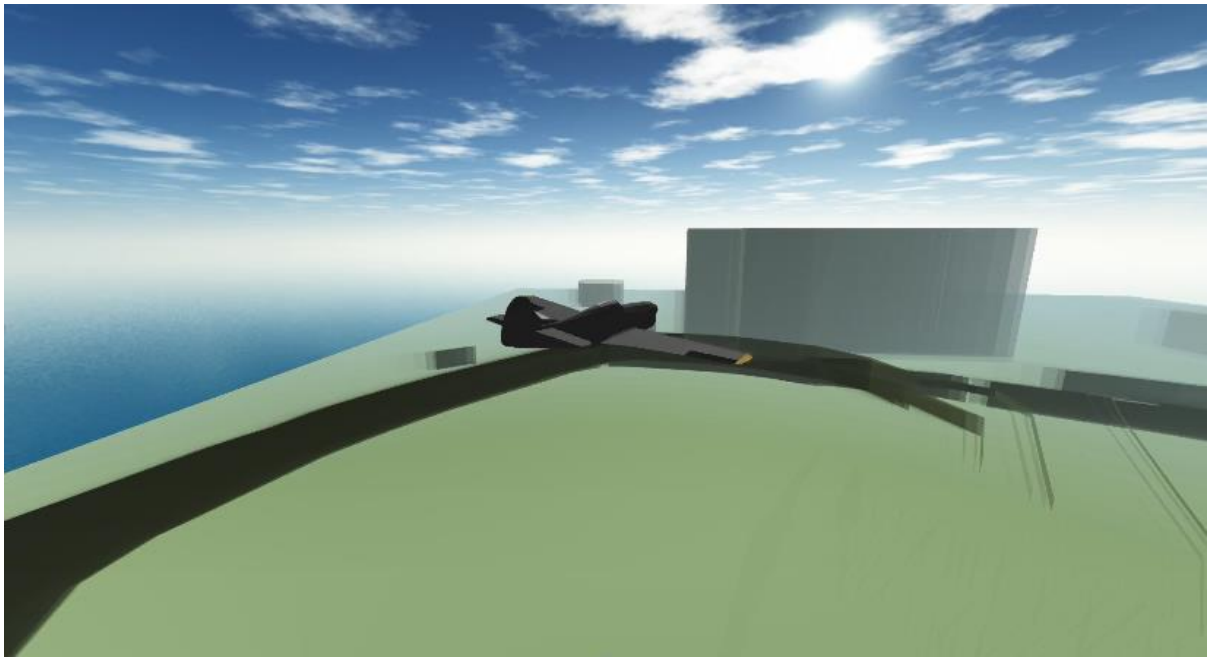


SHADOW MAP

FOG WEATHER



Final Effect



Improvement

Tongji Model

The model is fine and well designed. There are some good features & bad features of this model system.

a. We succeed to load the model into the memory & display it in the screen. But due to the incorrect process, we failed to load the whole model into the system. We incorrectly rasterize all the surface we read and then load it into memory, which leads to an overflow of the memory. The correct way should only rasterize the face that we can see instead of all the faces.

b. The shading of the model is not very well designed. The system can not load the shading of the lake & only treat the lake as the building so we cannot tell whether it's a lake or a building.

Aircraft Modeling System

The Aircraft model is doing great in our system. We do accomplish a lot of features as we designed at first.

a. We managed to load the OBJ file correctly & accomplish the shading work. Besides we do manage to control the airplane to go upstairs, downstairs, left , right and rotate with the keyboard. And we managed to zoom in and out with the ctrl keyboard.

b. But we still need more improvement. For instance, we did not achieve the crash system of the aircraft. We do design to make the crash animation but failed to properly calculate when the crash should happen and when it should not.

Roles in Group

During the implementation of the project, team members will work together, and the division of labor is as follows:

DIVISION OF LABOR:

- 杨文翔: Shader Module
- 杨宇航: Terrain, Terrain Renderer
- 高帅军: Shadow Map Module
- 贾育文: Camera controller, Airplane controller
- 秦睿: Airplane Module
- 左心笛: Loading Airplane Module, Camera Module

DIVISION OF PROJECT:

- Camera module;
- Modeling (build the models of the teaching buildings and the aircraft with SketchUp, then import the 3D model and draw the 3D model in OpenGL);
- Weather system & Lighting, shadow, and texture settings;
- Keyboard and mouse functions to implement interaction;
- Function integration and debugging

REFERENCES

[1] Johanna Choumert-Nkolo,Pascale Combes Motel,Leonard Le Roux. Stacking up the ladder: A panel data analysis of Tanzanian household energy choices[J]. World Development,2019,115.

[2] Jiaji Wu,Long Deng,Anand Paul. 3D Terrain Real-time Rendering Method Based on CUDA-OpenGL Interoperability[J]. IETE Technical Review,2015,32(6).

[3] Donald Hearn, M. Pauline Baker, 2012, Computer Graphics with OpenGL Fourth Edition, Prentice Hall.

[4] Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane, 2013, OpenGL Programming Guide:the Official Guide to Learning OpenGL, Version 4.3, Eighth Edition, Addison-Wesley Professional.

[5] Jiaji Wu,Long Deng,Anand Paul. 3D Terrain Real-time Rendering Method Based on CUDA-OpenGL Interoperability[J]. IETE Technical Review,2015,32(6).

[6] opengl-tutorial [DB/OL] http://www.opengl-tutorial.org/

[7] How to use assimp to load files [EB/OL] https://gamedev.stackexchange.com/questions/48302/using-assimp-to-load-dae-files

[8] Virtual Terrain [DB/OL] http://vterrain.org/

[9] The algorithm of tutorial [DB/OL] http://ogldev.atspace.co.uk/www/tutorial04/tutorial04.html

[10] The information of material [DB/OL] https://learnopengl.com/Lighting/Materials

[11] The information of lighting-maps [DB/OL] https://learnopengl.com/Lighting/Lighting-maps

[12] An introduction of Ray-tracing [DB/OL] http://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/ray-tracing-practical-example

[13] How to import collada [EB/OL] https://help.sketchup.com/en/sketchup/importing-and-exporting-collada-files