XI'AN JIAOTONG-LIVERPOOL UNIVERSITY

西交利物浦大学

COURSE WORK SUBMISSION

Name (ID Number)	Xuening Wang 1929720
Programme	BSc Information and Computing Science
Module Title	Computer Graphics
Module Code	CPT205
Assignment Title	3D Modelling Project
Submission Deadline	2021.12.13
Lecturer Responsible	Yong Yue

I certify that:

- I have read and understood the University's definitions of COLLUSION and PLAGIARISM (available in the Student Handbook of Xi'an Jiaotong-Liverpool University).

With reference to these definitions, I certify that:

- I have not colluded with any other student in the preparation and production of this work;
- this document has been written solely by me and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web);
- where appropriate, I have provided an honest statement of the contributions made to my work by other people including technical and other support staff.

I understand that unauthorised collusion and the incorporation of material from other works without acknowledgement (plagiarism) are serious disciplinary offences.

Signature ……Xuening Wang…… Date ……*2021.12.13*……

Assignment 2: 3D Modelling Project

Introduction

In this project, computer graphics techniques and OpenGL functions are applied to produce a 3-dimensional scene of a "smart city". The source code is well-configured under MS VC++ and OpenGL environments (freeglut library).

Design & Features

This 3D smart city is specially designed to convey the future planning concepts of a ***Cyber Safe City***. The design was initially inspired by an article in iotsecurityfoundation.org [1], the original design is shown below:

Figure 1: Sample 3D Smart City [1]

- *Buildings*

  This program requires a lot of 3D objects. In order to save code space and make code more understandable, geometric primitives are re-defined to satisfy the modelling requirements and bind texture more effectively.

  For example, the function

  ```
  void quad(int a, int b, int c, int d, GLfloat size, int e)
  ```

  re-defines a quadrilateral (quad) with binding texture (texture[e]).

  Then when drawing 3D objects, these functions are called to generate cubes/cylinders with corresponding textures. In this part, the textures refer to the building surface and details such as window, door and banner.

  Also, for loops are used to repeat certain operations and save code space.

Figure 2: Buildings

- *Cars and Trucks*

Cars and trucks are designed as an important part of smart city. In this part, hierarchical modelling is applied. A car/truck consists of a chassis and 4 wheels.

For example, when drawing the car, the components are generated by functions like

```
void drawFullCylinder(GLfloat radius, GLfloat height)
```

and translated to specified position, finally encapsulated into function `void car()`.

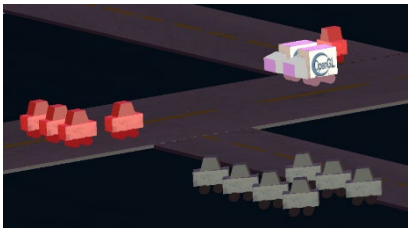


Figure 3: Cars and Trucks

- *Train, Bridge, Crossroad and Trees*

In this section, several objects in the city are designed. Predefined cylinders and cubes are used to generate 3D objects likewise in the previous parts.

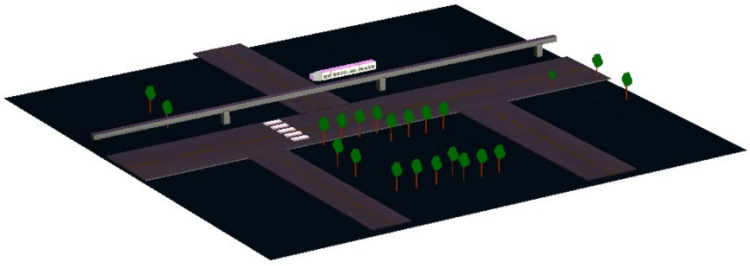


Figure 4: Train, Bridge, Crossroad and Trees

3D OpenGL Techniques

A series of OpenGL Techniques are used in this assignment:

Creation of Geometry

A great number of geometry primitives are created and re-defined in this assignment, including quads, polygons in 2D and cylinders, cubes in 3D.

Transformations

Different types of geometric transformation are applied, including scaling, translation, rotation and so on. For example, small objects need to be moved/adjusted to fit in the overall effects, and in this assignment, these are mostly realized by translation and scaling, wrapped by `glPushMatrix()` and `glPopMatrix()`.

Hierarchical Modelling

This program includes a number of geometry primitives, and these symbols are instanced using an instance transformation. Then, individual objects are grouped into a hierarchy that is represented by a tree structure.

For example, when designing the car, the Direct Acyclic Graph (DAG) could be represented like this:

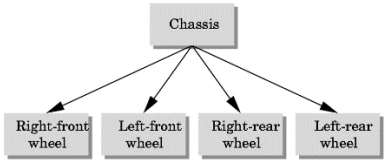


Figure 5: DAG of a Car

Meanwhile, the whole scene of the city could be represented like this:

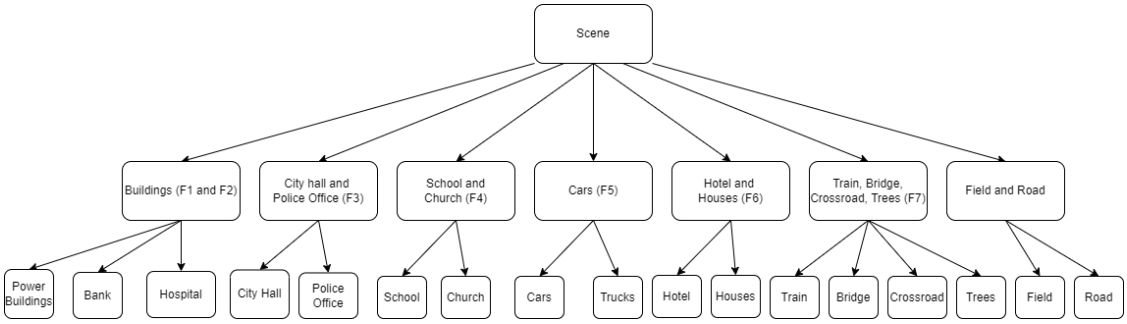


Figure 6: DAG of the city

Lighting and Materials

This program applies a number of spot lights and corresponding materials. When `glLight*()` is called to specify the position or the direction of a light source, the position or direction is transformed by the current model-view matrix. Meanwhile, a number of materials are used when drawing the objects.

Texture Mapping

A great number of textures are used in this program. The textures are bind corresponding predefined geometric primitives and used to generate the objects.

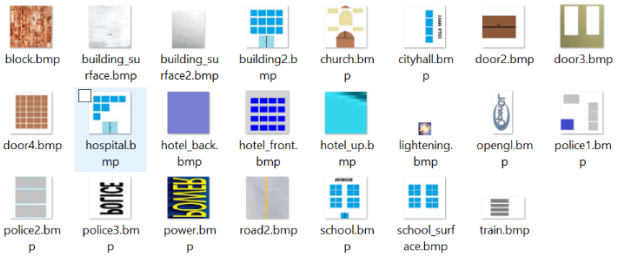


Figure 7: Textures used

Viewing and Projection

This program applies perspective projection, which appears more realistic since this is the way that human eyes and cameras form images. Also, this program enables user to change viewing position and direction, and zoom in/zoom out by mouse interaction, which will be mentioned in next section.

Animation and Interactions

A number of animation and interactions by mouse and keyboard are used in this assignment. Here are the following instructions (see ReadMe.md):

Keyboard Functions

Lighting

Press '0' to disable all lights

Press '1' to enable GL_LIGHT0

Press '2' to enable GL_LIGHT1

Press '3' to enable GL_LIGHT2

Fog

Press 'f' to increase fog density

Press 'g' to decrease fog density

Texture

Press 'b' to enable/disable textures

Exit

Press 'x' to exit the program

Mouse Functions

Press and drag GLUT_LEFT_BUTTON to change viewing position

Press and drag GLUT_RIGHT_BUTTON to zoom in/out

Press and drag GLUT_MIDDLE_BUTTON to change viewing direction

Special Key Functions

Press GLUT_KEY_UP to speed up the train

Press GLUT_KEY_DOWN to speed down the train

Press 'F1' to draw power building

Press 'F2' to draw hospital buildings

Press 'F3' to draw city hall and police office

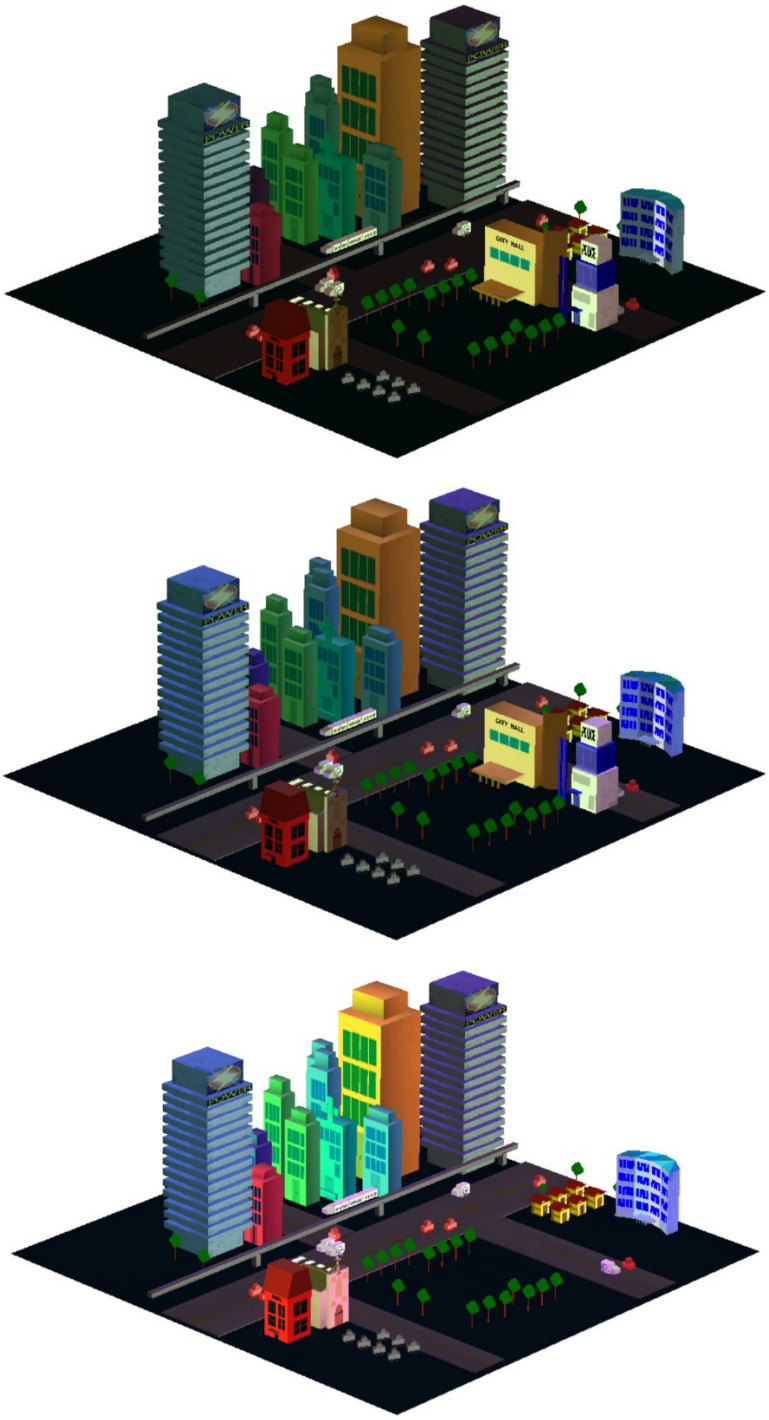
Press 'F4' to draw church and school

Press 'F5' to draw cars and trucks

Press 'F6' to draw hotel and houses

Press 'F7' to draw train, bridge, crossroad and trees

Sample Screenshots



With GL_LIGHT0, GL_LIGHT1 and GL_LIGHT2.

Reference

[1] A. Mihalic, Smart Cities -- The Emergence of the Cyber Safe Building, *IoT Security Foundation*, [Online]. Available: https://www.iotsecurityfoundation.org/smart_cities_the_emergence_of_the_cyber_safe_building/