Assesment 2: 3D Modelling Project

Module Title	Computer Graphics
Module Code	CPT205
Name	Penghao.Ouyang
Student ID	2141736
Degree Program :	Information and computing science

1. Introduction

The aim of this evaluation was to create a 3D scene containing a large number of static and dynamic objects. According to the requirements, this project uses only the "freeglut" library and the OpenGL environment to design a 3D scene. The theme of this project is to design a small house in a forest, and inside this scene there is also a beautiful lake and a viewing platform.

The design concept of the assignment is based on the theme of "Forest Hut" and contains many elements and graphics of the natural environment. The 3D scene for this assignment mainly contains a forest house, a lake and forest. Furthermore, we demonstrate the environment of the Forest Lodge at different times of the day and the appearance in different seasons. As winter sets in, the objects in the environment change. Finally, the project adds keyboard and mouse interactions taught in lab sessions, which can trigger different animations.

Several computer graphics techniques are used in this project to improve the visual results. One illustration is the use of Anti-Aliasing, which reduces graphical objects' jagged edges to give them a smoother, more realistic appearance [1].

2. Design and implementation

2.1 Creation of geometry

In order to better display realistic object effects in 3D scenes, this task uses the "glBegin()" function to draw basic 3D objects. In addition, specific shapes can be drawn by changing the parameters inside this function, such as using "GL_POLYGON" to draw polygons and "GL_QUADS" to draw quads. Through different combinations and matches, some prototypes of objects can be obtained.

In addition, this task also uses some functions that come with opengl to draw 3D graphics, such as "glutSolidSphere()" and "gluCylinder()". In order to better simplify the code and draw specific objects, this report uses custom methods such as "drawCylinder()", "drawBall()" and "drawCube()" to draw more complex graphics and objects.

2.2 Hierarchical Modelling

According to DeRose et al., the use of a layered modelling approach to progressively construct the scene and objects helps to improve rendering efficiency [2]. In the report that follows, I first present a general view of the effect, and then introduce the scene and objects progressively in order.

Environment: In this project, the whole environment consists of a cube. It is a cube with a texture pattern and the look of the texture changes with the seasons. Figure 1 presents the overall environment in winter. With the arrival of winter, the sky will be covered with snowflakes, the trees and the ground will be painted white by the snow, and the lake will be frozen. As the temperature rises in the spring, the snow and ice on the lake and trees melt and the forest takes on its original green appearance. Figure 2 shows how the environment looks in the spring.

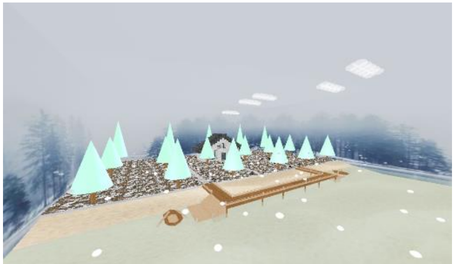

Figure 1: Winter

Figure 2: Spring

Lake: The whole environment consists of lake water and sand. To better present the real lake, the sand uses the 'glBegin()' function to set the slope, and the lake surface also sets a certain degree of transparency through 'glColor4f()', so that we can vaguely see the lake surface under the sand under the surface of the lake. Moreover, the lake surface will freeze in winter. Figures 3 and 4 show the corresponding scenarios.

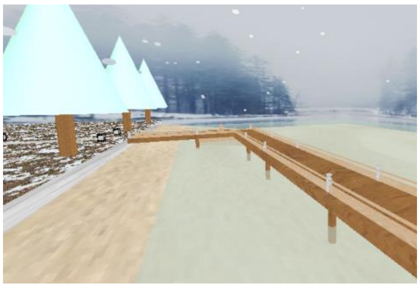

Figure 3: Lake(a)

Figure 4: Lake(b)

Platforms: This project draws platforms through custom functions 'drawCube()' and 'DrawBarrier()', and also maps objects to mimic the appearance of wooden platforms. Figure 5 shows the overall appearance of the platform. Figure 6 shows a wooden boat moored on either side of the platform, with the hull shape drawn by setting 'GL_POLYGON' and then the oars drawn by 'drawCube()'.

Figure 5: Platform

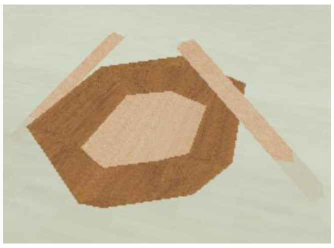

Figure 6: Boat

In order to make the platform visible at night, 'drawBall()' was used on the middle part of the fence to mimic a glow-in-the-dark bead. figure 6 and figure 7 show the fence at different times.

Figure 7: Barrier(a)

Figure 8: Barrier(b)

Land: Land is an area that contains several elements. Primarily, it is shaped as a rectangle with a grass texture covering the top of the rectangle. Next, panning the stone mapping with "glPushMatrix()" to create some sort of texture, and then using a for loop to form the stone path. The detailed rendering of the stone path and grass is shown in Figure 9 and Figure 10.

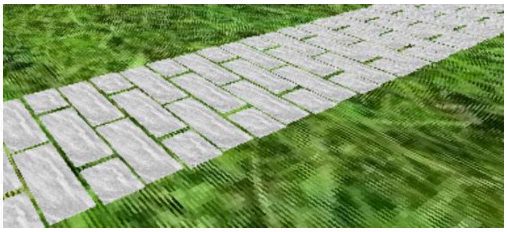

Figure 9: Road and grass(a)

Figure 10: Road and grass(b)

Street lamps and trees are distributed along the stone path. The trunks of the trees are mapped via 'drawCylinder()', and then their materials are defined via opengl's own ' gluCylinder() ' and using ' glMaterialfv ()' to define its material to mimic the leaves on the trunk. The street lamp was designed using hierarchical modelling to allow for overall movement. At night, the surface of the lamp is modelled to glow by defining the material of the lamp. Figure 11 and Figure 12 show the tree and the lamp respectively.

Figure 11: Lamp

Figure 12: Tree

House: The composition of the house is done with "drawCylinder()", "glPushMatrix()" and "drawCube()" in order to improve the readability of the code through layered modelling. The front of the house is made of glass and the transparency of the material is set so that the interior of the room can be seen. Additionally, I drew a door on the front of the house by mapping it. Figures 13, 14 and 15 show the front of the house at different times of the day.

Figure 13: House(a)

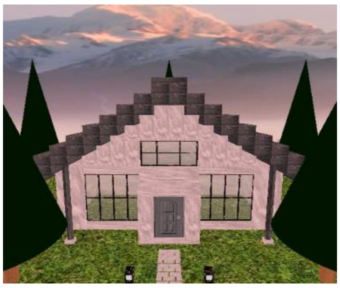

Figure 14: House(b)

Figure 15: House(c)

2.3 Transformations

This project uses three types of motion: rotate, translate and scale. This is achieved through the functions glRotatef(), glTranslatef() and glScalef(). In order to increase the simplicity of the code, almost all objects are modelled in the center, which greatly reduces the number of operations needed for rotation and translation. Besides, this report achieves the panning effect by setting the parameters inside the custom function.

2.4 Viewing and projection

In this project, we used the 'glFrustum()' function to construct the perspective shape and the near and far cropping planes, and the 'gluLookAt()' function to determine the position of the camera and the viewing point. Subsequently, we set the projection mode by calling the 'glMatrixMode(GL_PROJECTION)' function. This sequence of steps provides us with a perspective projection of the scene, ensuring an accurate representation of the visuals.

2.5 Lighting and materials

Lighting: There are three different types of light used in this project.

Ambient Light: There are three types of ambient light, which correspond to daytime, afternoon and nighttime light.
Spot Light: There are two types of spot light, corresponding to daytime and afternoon light. The daytime light is yellowish and shines from the upper right corner of the skybox, while the afternoon light is dark red and shines from the upper left corner.
Parallel light: The parallel light is white in color and is aimed to enhance the light intensity during the day, but the parallel light is disabled in the afternoon and evening.

Materials: The different likenesses of the objects are achieved by calling the 'glMaterialfv()' function and setting parameters such as 'GL_AMBIENT', 'GL_DIFFUSE,' 'GL_SHINESS', 'GL_SPECULAR', 'GL_EMISSION', etc. to achieve different appearances of objects. For example, by setting the transparency of WATER and GLASS to give them a real-world appearance. In addition, the materials of lamp and barrier are set to give them a glowing effect.

2.6 Texture mapping

This program uses texture mapping to enhance the realism of objects. Different wrapping and filtering effects can be achieved by adjusting the parameters of "glTexParameterf()". This report uses "GL_REPEAT" and "GL_CLAMP" to select the way the texture is rendered. In addition, I used two mapping methods: polyhedra are mapped using "glTexCoord2f()", while cylinders and spheres need to be texture mapped using the "GLUquadric" object.

3. Animation and interactions

3.1 Animation

Snow: The snowflakes in this experiment were drawn using the 'glutSolidSphere ()' function that comes with opengl, and the material was set to achieve a real-world snowflake appearance. figure 16 shows what a snowflake looks like.

Cloud: I drew the clouds by calling multiple 'glutSolidSphere ()' function, using the 'glColor4f()' to adjust the transparency of the clouds. Figure 17 shows the clouds.

Figure 16: Snow

Figure 17: Cloud

3.2 Interactions and instructions

Seasonal and time-of-day scenarios are shown in the second part of the report.

Keyboard: Press [c/C] to change the time and light.
Press [w/W], [s/S], [a/A], [d/D], to move back and forth to the left and right.
Press [j/J], [l/L], [i/I], [k/K] to turn your head to the left, right, up and down.

Mouse: Open the menu by clicking the right mouse button.
Change seasons by clicking the left mouse button.
Move up and down by sliding the mouse wheel up and down.

4. Conclusion

In this assignment, we used our knowledge of graphics to successfully construct a 3D scene featuring a cabin in the woods. Through careful design and modelling, we simulated morning, afternoon and evening as well as winter and spring scenes.

5. Reference

[1] P. P. Tanner, P. Jolicoeur, W. B. Cowan, K. Booth, and F. D. Fishman, “Antialiasing: A technique for smoothing jagged lines on a computer graphics image—an implementation on the Amiga,” Behavior Research Methods, Instruments & Computers, vol. 21, no. 1, pp. 59–66, Jan. 1989. doi:10.3758/bf03203871

[2] T. DeRose, M. Kass, and T. Truong, “Subdivision surfaces in character animation,” Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’98, Jul. 1998. doi:10.1145/280814.280826