CPT205-2223-S1-Computer Graphics

Shuchen Ji

2034172

Information and Computer Science

1 Design and functions

1.1 Design

In the assessment 2, I drew a three-dimensional amusement park with the sky, the grass, a Ferris wheel, a carousel, an enclosing wall, a road and some cars by using some relevant OpenGL libraries in freeglut. The sky, the grass, the carousel, the road and the enclosing wall were mapped with some pictures in bmp format by using texture mapping technology. In addition, an ambient lighting model also be implemented in this scene.

The Ferris wheel, the carousel and the cars all have animation effects. The Ferris wheel and the carousel are working and the keyboard can control their running speed and directions. The speed and directions of cars running on the road can also be switched by the keyboard. The colored lights on the Ferris wheel keep flashing with different colors which can also be stopped. The sky can switch between day and dusk with a different light intensity by mouse interaction. Moreover, the three-dimensional scene can change the perspective through the keyboard.

1.2 Functions

```
#define FREEGLUT_STATIC
#include <GL/freeglut.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <vector>
using namespace std;
int intWinWidth = 800; //Default window size
int intWinHeight = 600;
float fltFOV = 70; //Field Of View
float fltZoom = 1.0; //Zoom amount
float fltX0 = 0.0; //Camera position
float fltY0 = 150.0;
float fltZ0 = 500.0;
float fltXRef = 0.0; //Look At reference point
float fltYRef = 0.0;
float fltZRef = 0.0;
float fltXUp = 0.0; //Up vector
float fltYUp = 1.0;
float fltZUp = 0.0;
float fltViewingAngle = 0; //Used for rotating camera
const GLdouble PI = 3.1415926;
double cameraD = 10;
vector<GLubyte*> p;
GLuint texture[12];void when_in_mainloop(){ ... }
float ang = 0;//Ferris wheel rotate
float step = 0.1;
int time_interval = 10; // declare refresh interval in ms
void OnTimer(int value){ ... }
float dis1 = 0;
GLfloat stepd = 6;
int time_interval_CarRight = 35;
void OnTimer_CarRight(int value){ ... }
float dis2 = 0;
int time_interval_CarLeft = 35;
void OnTimer_CarLeft(int value){ ... }
```

```
58    int time_interval_Light = 200;
59    float step3 = 0.02;
60    float colorLight[] = { 0.2,0.1,0.2 };
61  ⊞void OnTimer_Light(int value) { ... }
76    float angCC = 0;//carousel rotate
77    float stepCCar = PI / 360;
78    int time_interval_CC = 80;
79  ⊞void OnTimer_CC(int value) { ... }
85  ⊞struct image { ... };
90    image loadTexture[10];
91  ⊞void ReadImage(const char path[256], GLint& imagewidth, GLint& imageheight, GLint& pixellength) { ... }
111 ⊞void myinit() { ... }
148   float dark = 0.0;
149 ⊞void light() { ... }
158   //two pillars
159 ⊞void DrawPillar1() { ... }
183   //map two pillars
184 ⊞void DrawPillarPic() { ... }
224   //enclosing wall
225 ⊞void DrawPillar2() { ... }
284   //map the wall
285 ⊞void wallPic0() { ... }
303   //front out
304 ⊞void wallpic1() { ... }
335   //behind in
336 ⊞void wallpic2() { ... }
363 ⊞void wallPic() { ... }
401   //small pillar
402 ⊞void DrawPillar3() { ... }
545 ⊞void DrawRail() { ... }
638   double ch1 = 0;
639   double ch2 = 0;
640   //map sky and grass
641 ⊞void DrawBackground() { ... }
682   //ferris wheel zhijia
683 ⊞void DrawZhijia() { ... }
```

```
752   //front lights
753 ⊞void DrawLight1() { ... }
780   //behind lights
781 ⊞void DrawLight2() { ... }
805   //pillar between light bars
806 ⊞void DrawSlash() { ... }
834   float colorCabin[3];
835   //ferris wheel cars
836 ⊞void DrawCabin(int i) { ... }
940 ⊞void DrawCabinRotate() { ... }
954   //the wheel
955 ⊞void DrawWheel() { ... }
1072  //map the road
1073⊞void roadPic() { ... }
1089⊞void drawTyre() { ... }
1098  //close to the screen-right direction
1099⊞void DrawTyresRight() { ... }
1124  //far away from the screen-left direction
1125⊞void DrawTyresLeft() { ... }
1150⊞void carRight1() { ... }
1179⊞void carRight2() { ... }
1197⊞void carRight3() { ... }
1254⊞void carRight4() { ... }
1271⊞void carRight5() { ... }
1300⊞void carRight6() { ... }
1357⊞void carLeft1() { ... }
1375⊞void carLeft2() { ... }
1394⊞void carLeft3() { ... }
1423⊞void carLeft4() { ... }
1441⊞void carLeft5() { ... }
1460⊞void carLeft6() { ... }
1489⊞void CarouselBottom() { ... }
1520⊞void CarouselPolar() { ... }
1541⊞void CarouselTop() { ... }
1559  float xc = -120;
1560  float zc = 0;
```

```
1561⊞void CarouselBody() { ... }
1589  //car on Carousel
1590⊞void carrr() { ... }
1651⊞void CarouselCar() { ... }
1667⊞void display() { ... }
1724⊞void displayObject() { ... }
1740⊞void reshapeWindow(GLint intNewWidth, GLint intNewHeight) { ... }
1744⊞void keyboard_input(unsigned char key, int x, int y) { ... }
1811⊞void mouse_input(int button, int state, int x, int y) { ... }
1823⊞int main(int argc, char** argv) { ... }
```

2 Interaction mode of mouse and keyboard

2.1 mouse

Click the left mouse button, the will switch from day to dusk.

Click the right mouse button, the will switch from dusk to day.

2.2 keyboard

Press a, the camera will turn left.

Press d, the camera will turn right.

Press w, the camera will move Up.

Press a, the camera will move down.

Press e, the camera will be zoomed in.

Press f, the camera will be zoomed out.

Press m, the Ferris wheel will accelerate rotation.

Press n, the Ferris wheel will slow down the rotation.

Press r, the rotation of the Ferris wheel will stop.

Press t, the rotation of the Ferris wheel will restart.

Press c, the Ferris wheel will change the rotation direction.

Press x, the rotation of the carousel will stop.

Press y, the rotation of the carousel will restart.

Press i, the cars will stop moving.

Press j, the cars will restart moving.

Press h, the cars will change the moving direction.

Press k, the lights will stop flashing.

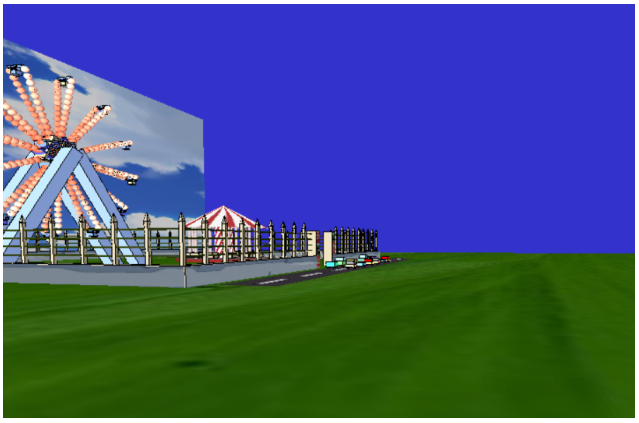
Press l, the lights will restart flashing.

Press q, the program will quit.

3 Screenshots

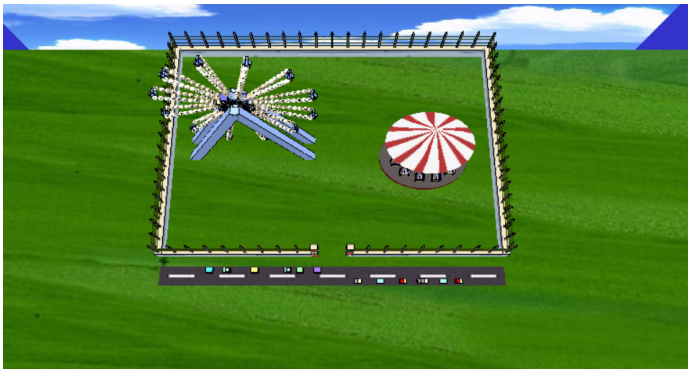
Main view:

Left view:



Right view:

Upward view:



Switch to dusk: