

Grade Form for 6GEN715 Project

Student Name: Lingfeng Zhang **Student Number:20152649**
Student Name: Lin Zhou **Student Number:20152652**
Student Name: Weichao Kong **Student Number:20152573**
Student Name: Chenxi Wang **Student Number:20152657**

Category	Items	Mark allocated	Grade
Demo (10 marks)	(H) house - A house should be created and displayed.	2	2
	(T) tree- At least one tree should be created and displayed	2	2
	(S) Sun – the sun is a must object.	2	2
	(E) extra environment object(s) - you can create any other extra object on the piece of land you have.	2	2
	(V) viewpoints - your program should enable viewer to view your home with different angles.	2	2
Sub-total:		10	10
Final Report (20 marks)	(C) clarity - Does your report explain everything that you did?	2	2
	(D) design - Did you design your implementation well? Did you use any CG technique(s) where appropriate?	2	2
	(E) English - Are there any spelling or grammatical errors? Is your writing clear and succinct?	2	2
	(N) neatness - Are the text, code and figures laid out well? Is your writing/printing legible?	2	2
	(T) thoroughness - Is the report complete? Did you leave important things out? Did you forget to describe important cases, rules, or program behaviour? Did you make errors in your project that aren't listed in the shortcomings?	2	2
Sub-total:		10	10
Total:		out of 20	

General Comments:

Dutys:

Lingfeng Zhang: The leader of our group whose responsibility is building the whole house, programming the code to simulate the rain animation and writing this report;

Lin Zhou: In charge of constructing the Environment which consist of colorful trees, the car and the land and learning how to append the textures on the models.

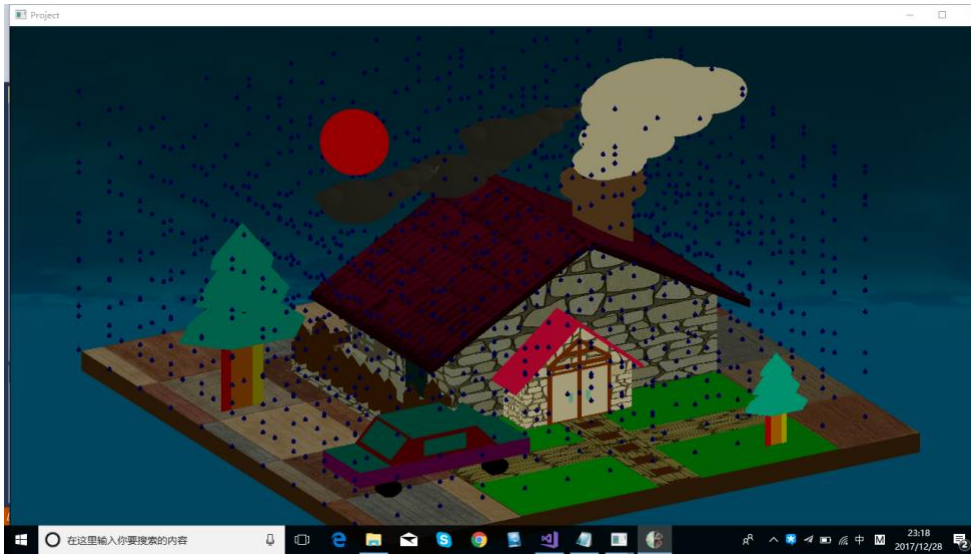
Weichao Kong: In charge of constructing the clouds and the sun and learning how to add the light in the coordinate system.

Chenxi Wang: the only female in our group, in charge of organising our objects in the whole world by calculating the coordinates, creating the model of raindrop and programming the function `specialKeys()` which we can use the keyboard to control the viewport and change the angles.

Introduction and Motivation

In this project we constructed the house with an open garden which includes the complex house, double colorful trees with different sizes, a luxury car, the sun and a few clouds. Firstly, the leader created a model of this project which was edited according to the lab5, which use the double buffer so that we can accelerate the drawing speed. On top of that, the reason why we use the same model to program our code is that we use the identical coordinate system so that we can organize the code together easier after everyone has finished his part of code. Secondly, our members draw the sketches of their models and then calculated the relative vertex positions. I think it's a complicated job because we must count the vertexes accurately. If some wrong graphics occurs, we have to recalculate them. It's a huge business. In this project, we faced plenty of questions such as software configuration bias and even we didn't know how to draw a circle and a sphere just using the vertexes rather than the glut function. To solve them, we searched for the solution on the Internet and dealt with it. (I'll describe this part in details at the third part of my report). The last thing is that we named our house as *The Harmonious House* because I think our house has a wonderful layout.

Design and Implementations:



As I said before, our model includes a lot of objects. So I'll introduce them in details as following:

-the house (most complex one): I used the `GL_QUAD_STRIP`, `GL_QUADS` and `GL_TRIANGLES` to create the outline of my house and used the white color. And then I still used the `GL_QUAD_STRIP` and `GL_QUADS` to set up the roof but the color is red. And I shrank the copy of the outline of house and the roof to become gate body, which includes some tricks. And I also used `GL_QUAD_STRIP` and `GL_QUADS` to create the window frames, door frames, balcony floor and the fence on the balcony. But I used the `GL_TRIANGLE_FAN` to create the top of fence bars. Because I have finished my door frames and window frames, I just need to add surfaces to the middle of frames but the windows and the door are still stereoscopic. I think it will confuse everyone's version. For chimney, it's hard to be realized. I searched for the information about how to draw a circle with OpenGL rather than the glut function on the Internet. The codes are following:

```
glBegin(GL_TRIANGLE_FAN);
for (int i = 0; i <= 360; i++)
    glVertex3f(radius * cos(2 * 3.14 / 360 * i), radius * sin(2 * 3.14 / 360 * i),
200.0);
```

`glEnd();`
 360 `GL_TRIANGLE_FAN`s consists of a circle. The formula is so abstract that I can't understand it. But I can modify it to create a cylinder (The idea comes from the `GL_QUADS`). The codes are following:

```
glBegin(GL_QUADS);
for (int i = 0; i <= 360; i++) {
```

```

    glVertex3f(radius * cos(2 * 3.14 / 360 * i), radius * sin(2 * 3.14 / 360 * i),
0);
    glVertex3f(radius * cos(2 * 3.14 / 360 * i), radius * sin(2 * 3.14 / 360 * i),
200);
    glVertex3f(radius * cos(2 * 3.14 / 360 * (i + 1)), radius * sin(2 * 3.14 /
360 * (i + 1)), 200);
    glVertex3f(radius * cos(2 * 3.14 / 360 * (i + 1)), radius * sin(2 * 3.14 /
360 * (i + 1)), 0);
}

```

The chimney consists of above two parts.

- the garden terrain: we just used GL_QUADS and GL_QUAD_STRIP with different colors to create the terrain and the road.

- double trees: we used GL_QUAD_STRIP with six colors to create a regular hexagonal prism with a gradient color and also used GL_POLYGON to create a regular hexagon with a gradient color. Both of them compose the tree trunk. And we used GL_TRIANGLE_FAN with two kinds of color to create the leafage. The first vertex used the dark green and the other vertexes used light green so that we can create the gradient green color from the top to the bottom. And we just used glScalef() function and glTranslatef() function to change the size and position of the copy of the first tree. It's our second tree. You should know the sum of the interior angles of a hexagon is 720 degrees, which will help you calculate the vertex position accurately.

- the car: we just used same way to create this object: GL_QUADS and GL_QUAD_STRIP for the body, GL_QUADS and GL_TRIANGLE_FAN for the tires (like the chimney).

- the sun: it's hard to draw because it's a sphere. Because drawing a sphere is a big job so we create a new self-defining function drawSphere():

```

void drawSphere(double r, int lats, int longs) {
    int i, j;
    for (i = 0; i <= lats; i++) {
        double lat0 = pi * (-0.5 + (double)(i - 1) / lats); // M_PI
        double z0 = sin(lat0);
        double zr0 = cos(lat0);

        double lat1 = pi * (-0.5 + (double)i / lats);
        double z1 = sin(lat1);
        double zr1 = cos(lat1);

        glBegin(GL_QUAD_STRIP);
        for (j = 0; j <= longs; j++) {
            double lng = 2 * pi * (double)(j - 1) / longs;
            double x = cos(lng);
            double y = sin(lng);

```

```

        glNormal3f(x * zr0, y * zr0, z0);
        glVertex3f(x * zr0, y * zr0, z0);
        glNormal3f(x * zr1, y * zr1, z1);
        glVertex3f(x * zr1, y * zr1, z1);
    }
    glEnd();
}
}

```

We can use this function to create a sun and change the radius according to the parameter *r*. The parameter *lat* defines how many horizontal lines you want to have in your sphere and *lon* how many vertical lines. Now there is a double iteration over *lat/lon* and the vertex coordinates are calculated. To be sure that the sphere is pure, you must use same number in parameter *lat* and *lon*.

-The clouds: it was made of many spheres which have different sizes and positions, using several times of function `drawSphere()`, `glTranslatef()` and `glScalef()`.

How to change the viewing angle?

We just add those codes after calling the function `glLoadIdentity()` each time:

```

glLoadIdentity();
glRotatef(rotate_x, 1.0, 0.0, 0.0);
glRotatef(rotate_y, 0.0, 1.0, 0.0);
glRotatef(rotate_z, 0.0, 0.0, 1.0);
glTranslatef(transfer_x, transfer_y, transfer_z);
glScalef(scale_change, scale_change, scale_change);

```

We can press the keyboard to rotate and move the model along the *x*, *y* and *z* axis and change the size of model. Just call the function `specialKeys()` in the function `main()`.

-rain animation: We just organized modified circle function which can create a circular cone and the sphere function. And we used three times loop to create thousands of raindrops. The animation is simple because we used random numbers to take the place of the parameter in the `glTranslatef()` function. The time is the parameter in the `srand()` function so that we could create different random numbers in different time. We can use the function `glutIdleFunc(display)` to draw our graphic infinitely.

-Shortcut keys: we add some codes just follow closely all `glLoadIdentity()`, `glTranslatef()`, `glRotatef()`, `glScalef()`, and add 8 Fn to control more different transformations.

-Light: We just added light at the origin of coordinate and used `glLightfv(GL_LIGHT0, GL_DIFFUSE, GLfloat array[])` to change the light color and set up a diffuse light. We also used `glLightfv(GL_LIGHT0, GL_POSITION, GLfloat array[])` to set up the light direction from the origin of coordinate. Light direction is a vector.

`glEnable(GL_LIGHTING)` is used to be sure whether we'll use the light and `glEnable(GL_LIGHT0)` to create a light.

-Texture: We could use the `glEnable(GL_TEXTURE_2D)` before we append the texture. After that, we must use `glDisable(GL_TEXTURE_2D)` to close the texture unless you want your texture to influence your object color. We must cite the path of the picture. It's an important thing that we must relate the texture coordinates to the objects coordinates using this form:
`glTexCoord2f(x, y); glVertex3f(x, y, z);`
and we must add the picture into the appropriate folder.

We can move into this house just press F3 with right angle and we'll see another picture in the house. It's surprise!



Shortcomings:

Although we add the light on this coordinate system, the effect is not perfect. The whole color of our model tends to be dark because we don't know how to remove the light position. Although we appended the texture and set up the light, we also have a lot of questions about how the function worked. And we also don't know how to append the texture to the sphere and circle. During we were finishing this project, the object color have a great impact on the texture color. And we can't solve it properly. Besides, we also found that the texture will be appended on both sides. We don't know how to control it except creating a new surface with this texture and just locating it extremely near to the aimed surface, which will confuse the human version. And we also want use the 24 alphabet to set up the shortcut keys but we failed. We just can use Fn to set up more shortcuts. When we create the sphere, its scale will be influenced by the window aspect ratio. We must modulate the size using `glScalef()`. We also want to let the clouds and smog move but we have no any time to do that. Because the time is limited, so we

have taken these problems to send this report. But we will find the reasons about them on the Internet or other kinds of methods and continue to consummate our project to be sure we totally acquire these knowledges.

Conclusions:

All in all, our group have finished this project. There is no denying that this project is a huge one. We have learned a lot of knowledge about the OpenGL programming and have acquired the knowledge about the basic principle of the coordinate system operation. After that, we know how to draw points, lines, faces and stereo shapes. We also know how to append the texture on the surface of the objects and set up the lights in the coordinate system to let our model look like more natural. We must add the function `glLoadIdentity()` before the first `glBegin()` or after the last `glEnd()` because the new graph will be draw based on the last coordinate system. If we don't do that, the coordinates of the first model will be confused. It's a weird situation that our tree trunk was squashed flat although our coordinates are totally correct. Our group members cooperate so tacit that we can finish our project well in a short time.

We also cited some codes from the examples which were given by our professor. Thank you professor!

Appendix:

We set up 12 shortcut keys:

`GLUT_KEY_RIGHT` //右键, 沿着x轴正方向转5度, 转的方向右手螺旋定则
`GLUT_KEY_LEFT` //左键, 沿着x轴逆方向转5度
`GLUT_KEY_UP` //上键, 沿着y轴正方向转5度
`GLUT_KEY_DOWN` //下键, 沿着y轴逆方向转5度
`GLUT_KEY_F1` //F1 感觉人右移动
`GLUT_KEY_F2` //F2 感觉人左移动
`GLUT_KEY_F3` //F3前进, 相当于放大然后前进
`GLUT_KEY_F4` //F4后退相当于缩小然后后退
`GLUT_KEY_F5` //F5沿着z轴正方向转5度
`GLUT_KEY_F6` //F6沿着z轴逆方向转5度
`GLUT_KEY_F7` //F7沿着y轴正向移动5个单位
`GLUT_KEY_F8` //F8 沿着 y 轴逆向移动 5 个单位

References:

\Slides\14-Lighting-n-Shading\Examples\Lighting-02-sphere-with-light
\Slides\14-Lighting-n-Shading\Examples\Lighting03-Phong\Lighting03-Phong
\Slides\15-TextureMapping\examples\04-TurningCube
\Labs\Lab05-Multiple-Objects-in-World\ArrowControlledCube_main.cpp
<https://zhidao.baidu.com/question/1957329110249011660.html>

If you want to open our project, you just open the folder 'project' and double click the `TextureMapping5.sln` because we added our codes in this project.