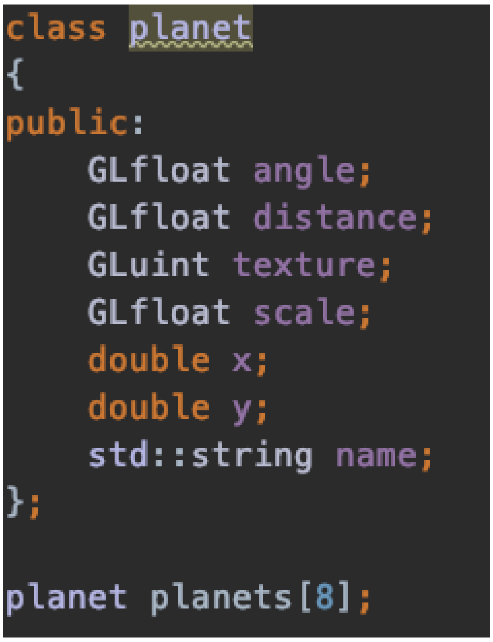
Sami Emre Erdogan
2019280513

Assignment 3 Solar System

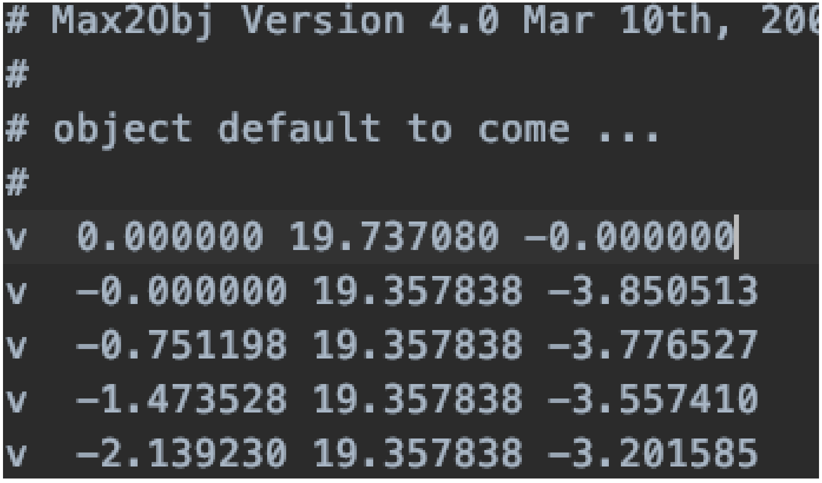
First, I had to create a class for the planet and define its properties. Such as angle, how much distance does the planet have to the sun, texture id, scale size, x and y coordinates on the GL window, their name etc.



```
class planet
{
public:
    GLfloat angle;
    GLfloat distance;
    GLuint texture;
    GLfloat scale;
    double x;
    double y;
    std::string name;
};

planet planets[8];
```

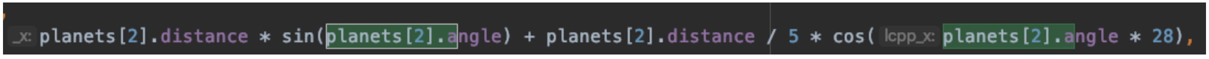
I tried to manually code the sphere but it was too difficult for me. Then I tired GlutSphere function to create a planet however I wasn't really familiar with it. Then I thought of the first mesh assignment that we had. What if I used an existing planet.obj file instead and it worked perfectly. Screenshot of the file:



```
# Max2Obj Version 4.0 Mar 10th, 20
#
# object default to come ...
#
v  0.000000 19.737080 -0.000000
v  -0.000000 19.357838 -3.850513
v  -0.751198 19.357838 -3.776527
v  -1.473528 19.357838 -3.557410
v  -2.139230 19.357838 -3.201585
```

Sami Emre Erdogan
2019280513

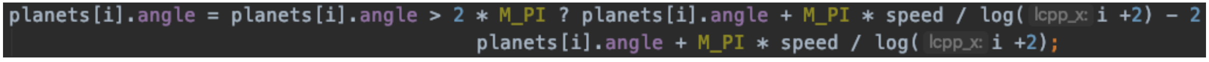
The angle and distance of the each planet has been calculated as follows although I really don’t know the appropriate values I tried to make it as realistic as possible



```
_x: planets[2].distance * sin(planets[2].angle) + planets[2].distance / 5 * cos( lcpp_x: planets[2].angle * 28),
```

I wanted to show the example of the moon as we have to make the calculations according to planets[2] which is planet earth because the moon has to go around the moon.
The other planets will be calculated respected to the sun.

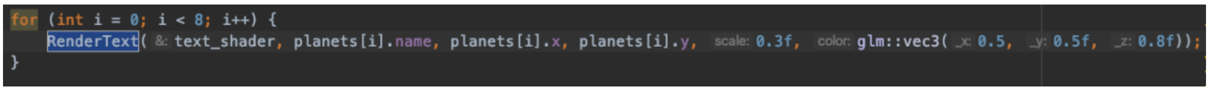
The angle of the planet has been calculated as follows:



```
planets[i].angle = planets[i].angle > 2 * M_PI ? planets[i].angle + M_PI * speed / log( lcpp_x: i +2) - 2
                                                 planets[i].angle + M_PI * speed / log( lcpp_x: i +2);
```

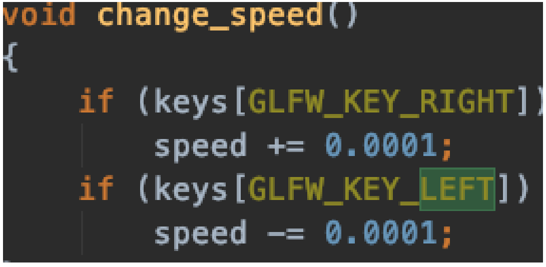
I also wanted to talk about is the text.

RenderText function was responsible of rendering the text.
the glhProjectf function was responsible of displaying the planet names under each planet by obtaining the x y coordinates form the objects. Multiplying with the vertices will give us the allocation to the respective planet x y coordinates.



```
for (int i = 0; i < 8; i++) {
    RenderText( &: text_shader, planets[i].name, planets[i].x, planets[i].y, scale: 0.3f, color: glm::vec3( _x: 0.5, _y: 0.5f, _z: 0.8f));
}
```

Lastly I want to mention that I have created left arrow , right arrow to control the rotation speed around the sun and the moon around the earth respectively.



```
void change_speed()
{
    if (keys[GLFW_KEY_RIGHT])
        speed += 0.0001;
    if (keys[GLFW_KEY_LEFT])
        speed -= 0.0001;
```

The movement of the camera is based on our tutorials with camera.h by using w a s d keys.