Final Project

Name: Chun Han Chen

Email: chenchu2@oregonstate.edu

1. Proposal

In this final project, I would like to do a solar system. Also, I will set a view and it could watch the constellation which is Scorpio from the Earth.

For the solar system, I plan to use texture mapping for all planets and the lighting for the sun. And adjust the distance and orbit period of all planets in proportion. Considering the background, I plan to use galaxy texture or something like this without the whole black background.

For the Scorpio, I plan to set up the inside view and outside view so that Scorpio can be seen when the user is in the inside view from the Earth. And will use animation to connect these stars with lines to make it easy for users to watch. (Fig 1.)

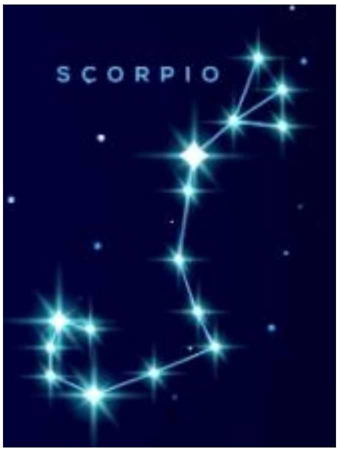


Fig 1. Scorpio

2. What you actually did for your project, with images

I did 10 planets in total. Respectively are the Sun, Mercury, Venus, Moon, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune from the center to the outer. Meanwhile, I also did Saturn’s ring and Uranus’s ring. In the background part, I changed the black background to the Milky Way to make the result more like the solar system.

I adjust the planet diameter and planet orbital radius in proportion to each planet. Although it is not very accurate, it is still close to a certain value according to Kepler's Third law of planetary motion. The reason for the inaccuracy will be described below.

In the center of the sun, I placed a light source point which is point light. This can illuminate every planet facing the sun.

Finally, I did a Scorpio. I set up inside and outside. When you in outside, you can not see the Scorpio. You only can see the Scorpio when you in the inside. Also, I did point light on the Scorpio.

3. How your project differs from what you proposed, and why

The first is the part of planet diameter and planet orbital radius. Excluding the sun does not count, because the sun is too huge, which will cause the output screen to be insufficient. In addition, Jupiter and Saturn are also quite huge. If I keep everything to scale, then the output screen is not enough, or Mercury, the Moon, and other smaller planets will be almost invisible. Therefore, I only slightly adjusted the size of Jupiter and Saturn, and then the Sun must be the largest of the 10. In this way, planet orbital radius makes sense. I have tried my best to make all planets follow Kepler's Third Law of planetary motion.

Next is the Scorpio part, which is a bit different from what I expected, but I still tried my best to make it. I expect to be able to watch Scorpio on the earth, but the earth will continue to rotate, so the procedure will become more complicated. So, I put my perspective on the sun and then watch Scorpio.

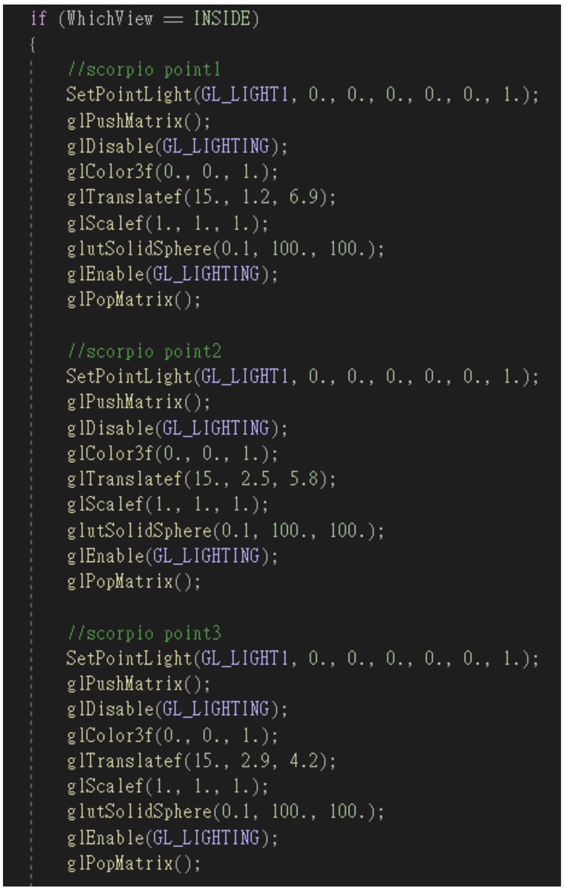
4. What you learned from doing this project (i.e., what you know now that you didn't know when you started)

The first one is that when the revolution and rotation were created at the beginning, a problem arose that I could only revolve the planet but not rotation. Later I realized that the professor in the first class said that the OpenGL program was read from bottom to top. So, I changed the order. That is, first let the planet rotate, then let the planet translate, and finally let the planet rotate once, so as to achieve rotation and revolution.

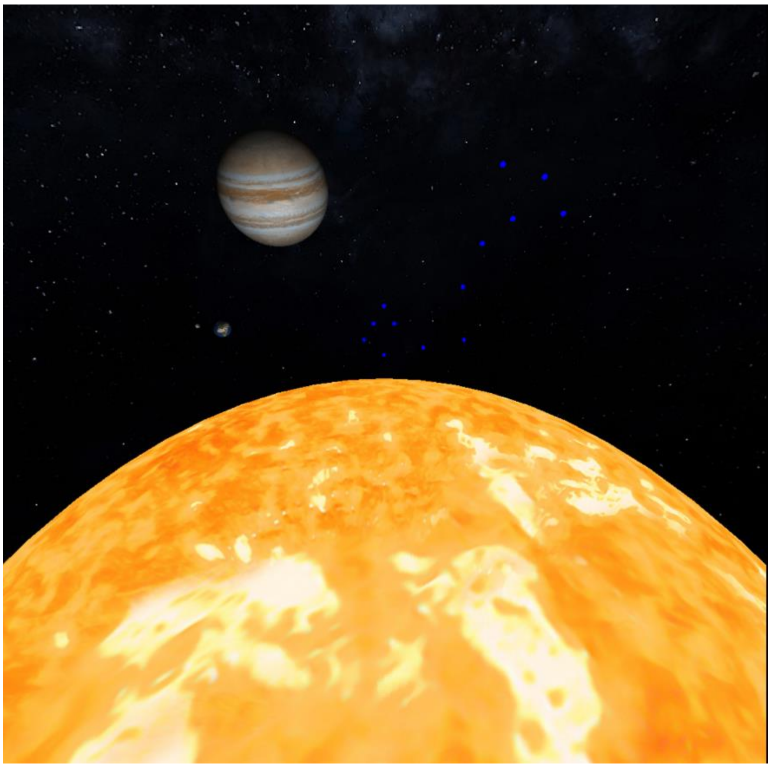
The second one is I want to try to replace the background. But at first, I did not know how to proceed. Later, I tried to use a very large sphere and then apply the texture, so that I could successfully change the background.

6. Some images that are especially representative of what you did

1.

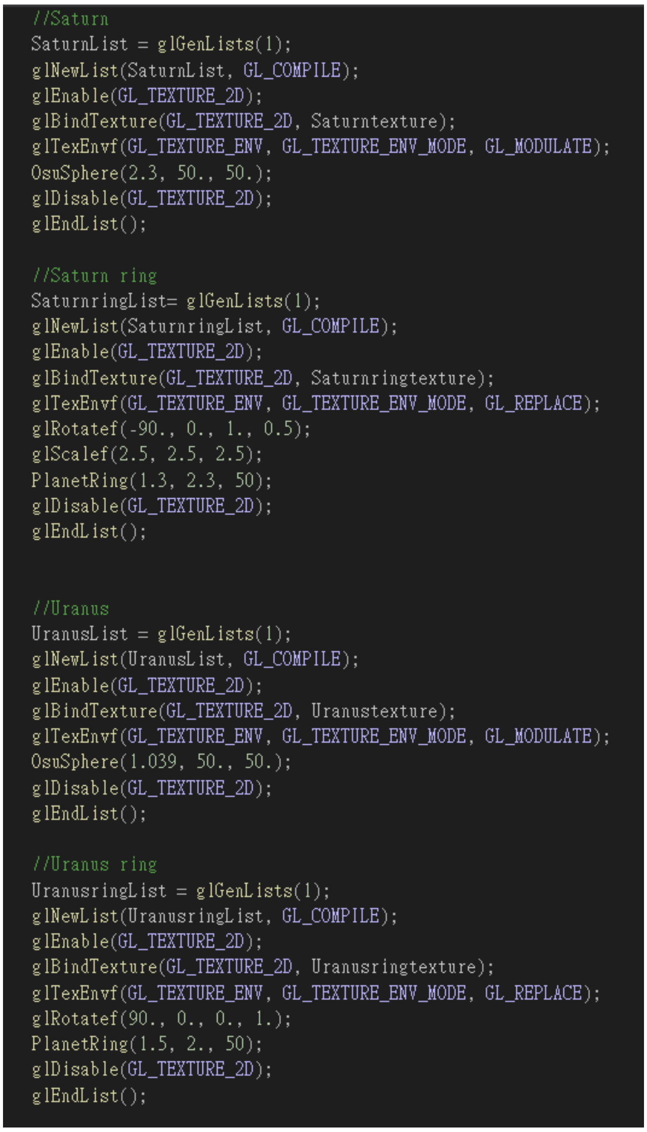


```
if (WhichView == INSIDE)
{
    //scorpio point1
    SetPointLight(GL_LIGHT1, 0., 0., 0., 0., 0., 1.);
    glPushMatrix();
    glDisable(GL_LIGHTING);
    glColor3f(0., 0., 1.);
    glTranslatef(15., 1.2, 6.9);
    glScalef(1., 1., 1.);
    glutSolidSphere(0.1, 100., 100.);
    glEnable(GL_LIGHTING);
    glPopMatrix();

    //scorpio point2
    SetPointLight(GL_LIGHT1, 0., 0., 0., 0., 0., 1.);
    glPushMatrix();
    glDisable(GL_LIGHTING);
    glColor3f(0., 0., 1.);
    glTranslatef(15., 2.5, 5.8);
    glScalef(1., 1., 1.);
    glutSolidSphere(0.1, 100., 100.);
    glEnable(GL_LIGHTING);
    glPopMatrix();

    //scorpio point3
    SetPointLight(GL_LIGHT1, 0., 0., 0., 0., 0., 1.);
    glPushMatrix();
    glDisable(GL_LIGHTING);
    glColor3f(0., 0., 1.);
    glTranslatef(15., 2.9, 4.2);
    glScalef(1., 1., 1.);
    glutSolidSphere(0.1, 100., 100.);
    glEnable(GL_LIGHTING);
    glPopMatrix();
```

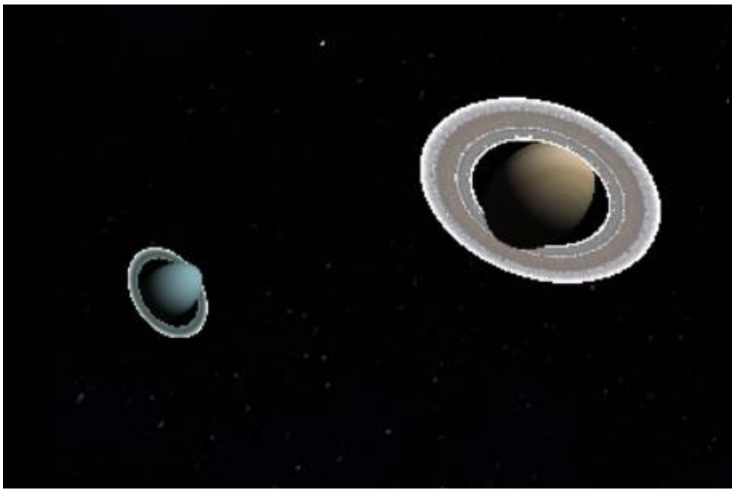


This is Scorpio. I set 13 point lights to creat Scorpio.

2.

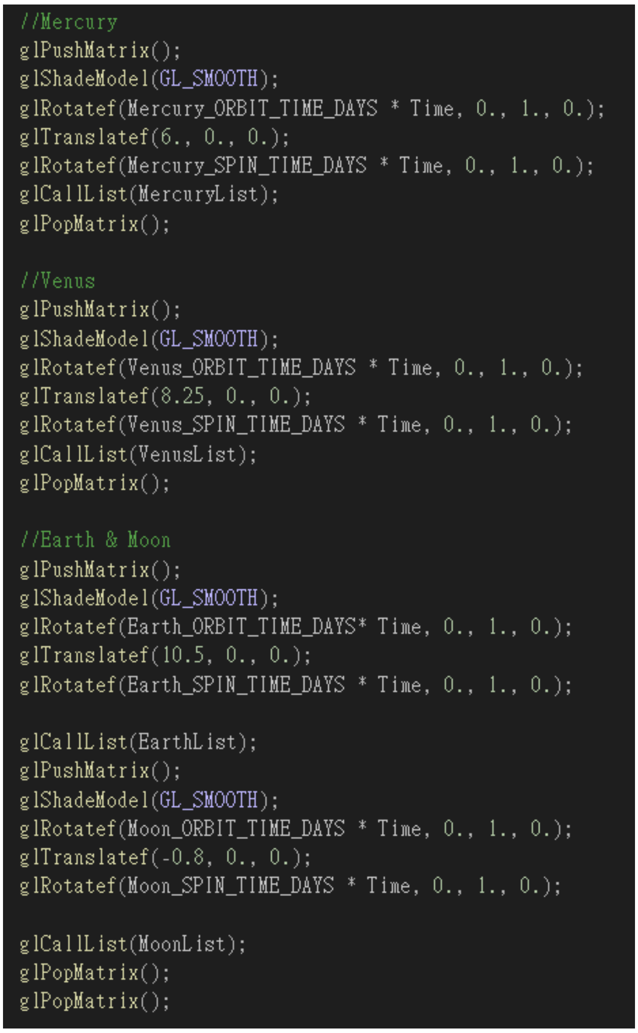


```
//Saturn
SaturnList = glGenLists(1);
glNewList(SaturnList, GL_COMPILE);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, Saturntexture);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
OsuSphere(2.3, 50., 50.);
glDisable(GL_TEXTURE_2D);
glEndList();

//Saturn ring
SaturnringList= glGenLists(1);
glNewList(SaturnringList, GL_COMPILE);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, Saturnringtexture);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
glRotatef(-90., 0., 1., 0.5);
glScalef(2.5, 2.5, 2.5);
PlanetRing(1.3, 2.3, 50);
glDisable(GL_TEXTURE_2D);
glEndList();


//Uranus
UranusList = glGenLists(1);
glNewList(UranusList, GL_COMPILE);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, Uranustexture);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
OsuSphere(1.039, 50., 50.);
glDisable(GL_TEXTURE_2D);
glEndList();

//Uranus ring
UranusringList = glGenLists(1);
glNewList(UranusringList, GL_COMPILE);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, Uranusringtexture);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
glRotatef(90., 0., 0., 1.);
PlanetRing(1.5, 2., 50);
glDisable(GL_TEXTURE_2D);
glEndList();
```

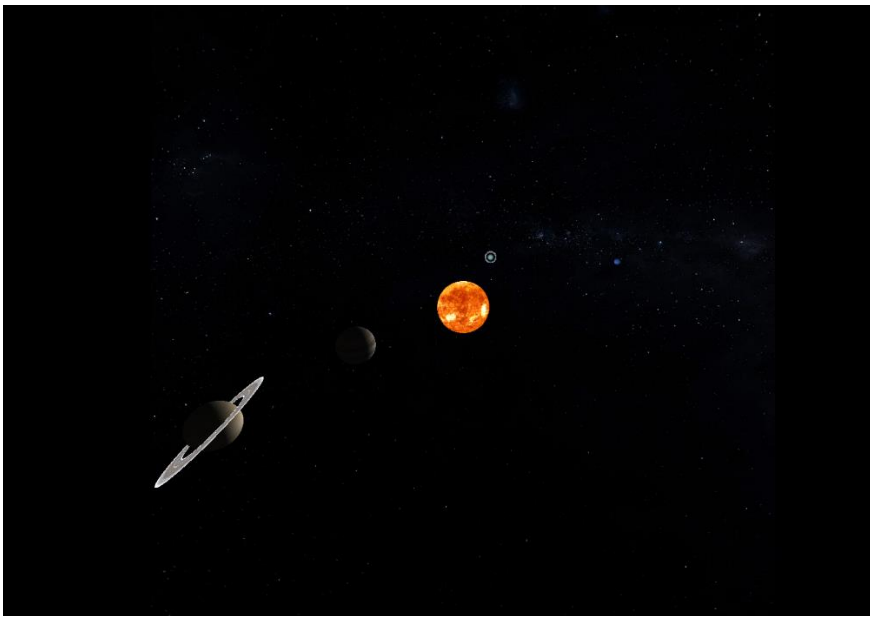


This is Saturn’s ring and Uranus’s ring function.

3.



```
//Mercury
glPushMatrix();
glShadeModel(GL_SMOOTH);
glRotatef(Mercury_ORBIT_TIME_DAYS * Time, 0., 1., 0.);
glTranslatef(6., 0., 0.);
glRotatef(Mercury_SPIN_TIME_DAYS * Time, 0., 1., 0.);
glCallList(MercuryList);
glPopMatrix();

//Venus
glPushMatrix();
glShadeModel(GL_SMOOTH);
glRotatef(Venus_ORBIT_TIME_DAYS * Time, 0., 1., 0.);
glTranslatef(8.25, 0., 0.);
glRotatef(Venus_SPIN_TIME_DAYS * Time, 0., 1., 0.);
glCallList(VenusList);
glPopMatrix();

//Earth & Moon
glPushMatrix();
glShadeModel(GL_SMOOTH);
glRotatef(Earth_ORBIT_TIME_DAYS* Time, 0., 1., 0.);
glTranslatef(10.5, 0., 0.);
glRotatef(Earth_SPIN_TIME_DAYS * Time, 0., 1., 0.);

glCallList(EarthList);
glPushMatrix();
glShadeModel(GL_SMOOTH);
glRotatef(Moon_ORBIT_TIME_DAYS * Time, 0., 1., 0.);
glTranslatef(-0.8, 0., 0.);
glRotatef(Moon_SPIN_TIME_DAYS * Time, 0., 1., 0.);

glCallList(MoonList);
glPopMatrix();
glPopMatrix();
```



This is solar system and all planet function are in display().

7. Link

https://media.oregonstate.edu/media/t/1_gti6vhxt