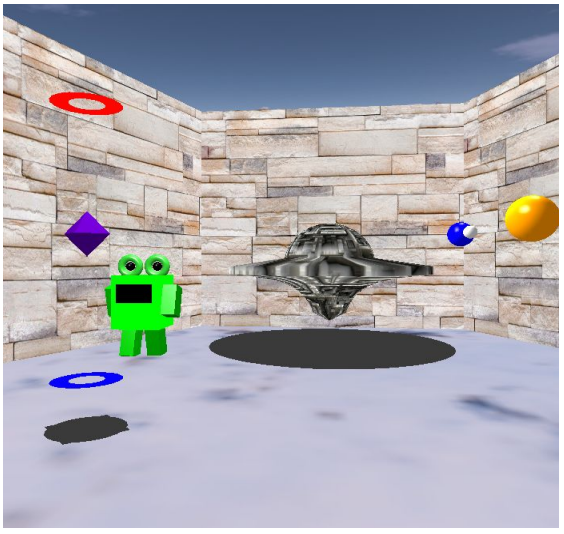
COSC363 Assignment1
Report (2020)

Name: Canying Liu
Student Number: 47108377

Brief Description

The scene shows a museum in the shape of an irregular hexagonal prism located in the Antarctic Observation Centre. There is a UFO model, a planetary orbit model, and a magnetic levitation model exhibited in the museum and a robotic rubbish bin is hanging around collecting rubbish.

Entrance



An inside look

UFO model

A textured UFO is generated by sweep surfaces, using mathematical equations. It is rotating on its y-axis.

Planetary orbit model

The planetary orbit model shows the orbits of planets including the Sun(orange ball), the Earth(blue ball), and the Moon(white ball). All of them are rotating on their axis by a certain degree depending on the different orbital periods. The Moon is in orbit around the Earth, where the Earth circulars the Sun.

Magnetic levitation model

The magnetic levitation model includes two discs and an octahedron. The blue disc represents the South magnetic pole, and the red disc indicates the North magnetic pole. The

magnetic octahedron is floating between these two discs and will suspend at each end for a short period.

Robotic rubbish bin

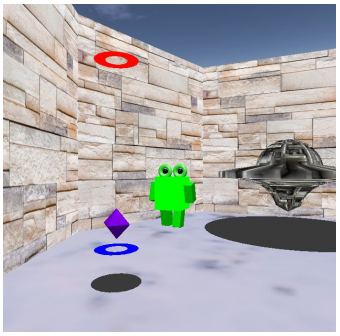
The robotic rubbish bin is composed of six cubes of different sizes, two torus, and two spheres. It has limbs, eyes, and a mouth, and it walks around the museum.

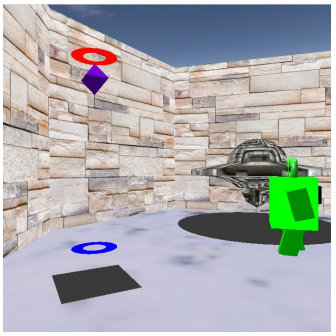
Extra Feature Implemented

Planar shadows

There are three planar shadows cast by spotlights in the scene: a circular shadow of the UFO, a square shadow of the magnetic octahedron, and a disc shadow of the South pole.

Spotlight on moving objects



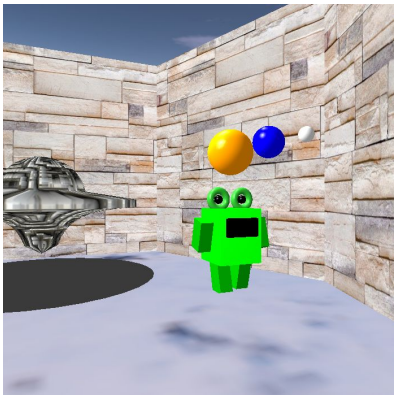


The shadow of the magnetic levitation model is changing with the movement of the octahedron. When the octahedron approaches the light source, its shadow gets larger and covers the shadow of the South pole disc, leaving a square shadow on the floor. However, a circular shadow is casting when heading in the opposite direction.

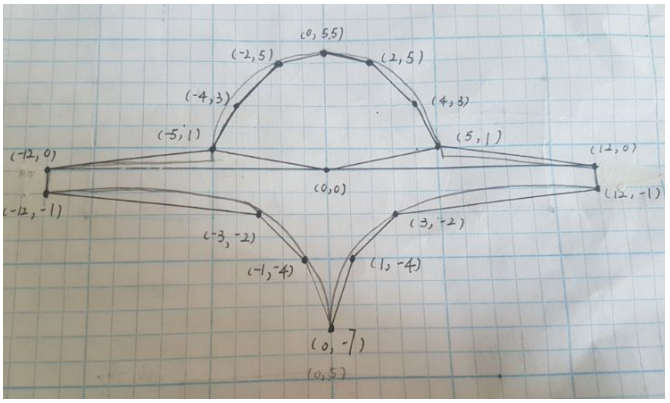
Physics based animation

The Sun rotates around its axis(which is titled by 7.25° from perpendicular to the ecliptic) and has an orbital period of 24 days. The Earth is self-rotating on its axis(23.5°) with a period of 1 day and is in orbit around the Sun with an orbital period of 365 days. The Moon rotates on its axis(6.68°) and orbits around the Earth with a period of 27 days. According to the information, the Moon will finish 12 complete orbits around the Earth when the Earth orbits the Sun once. At the same time, the Sun will rotate on its axis roughly 15 times.

```
void planets()
{
    glColor3f(1., 0.549, 0.);              //the Sun
    glPushMatrix();
        glTranslatef(7, 5, 7);
        glRotatef(15*rotation_angle*7.25, 0, 1, 0);
        glutSolidSphere(0.7, 30, 30);
    glPopMatrix();

    glColor3f(0., 0., 1);                  //the Earth
        glPushMatrix();
        glTranslatef(earth_x, 5, earth_z);
        glRotatef(rotation_angle*23.4/3, 0, 1, 0);
        glutSolidSphere(0.4, 30, 30);
    glPopMatrix();

    glColor3f(0.863, 0.863, 0.863);        //the Moon
        glPushMatrix();
        glTranslatef(earth_x+cos(-12*rotation_angle), 5, earth_z+sin(-12*rotation_angle));
        glRotatef(12*rotation_angle*6.68, 0, 1, 0);
        glutSolidSphere(0.2, 30, 30);
    glPopMatrix();
}
```

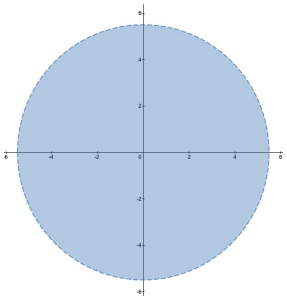


Sweep surfaces

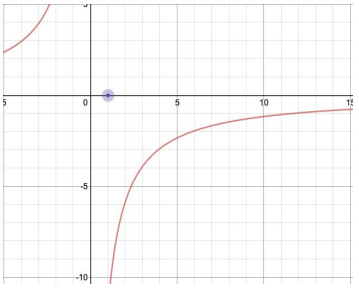



```
//----------------------Mesume Exhibits Structures----------------------
//================================================================
//--------------------UFO Structure(Sweep Surfaces)--------------------
void ufo(int shadow)
{
    if (shadow == 1) {                          //is the shadow
        glColor4f(0.2, 0.2, 0.2, 1.);
    } else {
        glEnable(GL_TEXTURE_2D);
        glBindTexture(GL_TEXTURE_2D, txId[7]);
    }

    const int N = 61;
    float vx_init[N] = {0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8,
        2, 2.2, 2.4, 2.6, 2.8, 3, 3.2, 3.4, 3.6, 3.8, 4, 4.2, 4.4, 4.6,
        4.8, 5, 5.2, 5.4, 12, 12, 11, 10, 9, 8, 7, 6.8, 6.6, 6.4, 6.2,
        6, 5.8, 5.6, 5.4, 5.2, 5, 4.8, 4.6, 4.4, 4.2, 4, 3.8, 3.6, 3.4,
        3.2, 3, 2.8, 2.6, 2.4, 2.2, 2, 0};
    float vy_init[N] = {5.5, 5.469, 5.485, 5.467, 5.442, 5.408, 5.367,
        5.319, 5.262, 5.197, 5.123, 5.041, 4.949, 4.847, 4.734, 4.61,
        4.473, 4.323, 4.158, 3.976, 3.775, 3.551, 3.3, 3.015, 2.685,
        2.291, 1.792, 1.044, 0, -0.98, -1.07, -1.176, -1.307, -1.471,
        -1.681, -1.73, -1.783, -1.838, -1.898, -1.961, -2.028, -2.101,
        -2.179, -2.262, -2.353, -2.451, -2.558, -2.674, -2.801, -2.941,
        -3.096, -3.268, -3.46, -3.676, -3.922, -4.202, -4.525, -4.902,
        -5.348, -5.882, -7};
    float vz_init[N]= {0};

    float vx[N], vy[N], vz[N];
    float wx[N], wy[N], wz[N];
```

The UFO model is constructed by sweep surfaces, and the curve is generated by two mathematical equations. A sketch above shows the design of the model. Points on the semicircle are obtained by the equation one (x-a)^2+(y-b)^2<r where a,b = 0 and r = 5.5. Points with x between 12 and 2 inclusive and with y in range -0.98 through -5.882 inclusive are plotted using equation two y = 1/(-ax) with a = 0.085.



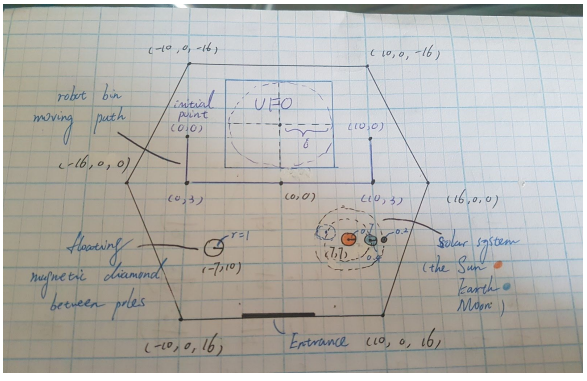
equation one



equation two

Collision detection

All models have been put in a proper place without interacting with each other. Any animations in the scene are limited in their scopes and will not collide with the walls. The camera could not reach the outside of the skybox, and it will be bounced off the skybox walls if it has a collision.



SkyBox

A 50*50 skybox is textured with different .tga files. There is no gap between each side and the mapped walls are connected. The border of the skybox is set, and the camera can only move within the skybox.

Control Functions

Up Arrow Key: Move the camera a step forward.
Down Arrow Key: Move the camera a step backward.
Left Arrow Key: Change the direction of the camera towards left by 5.7 degrees.
Right Arrow Key: Change the direction of the camera towards right by 5.7 degrees.

Build Commands

The .cpp file is built using Geany under the Oracle Virtual Box.

Method1:

Open terminal, cd to the project directory, and enter the following commands.

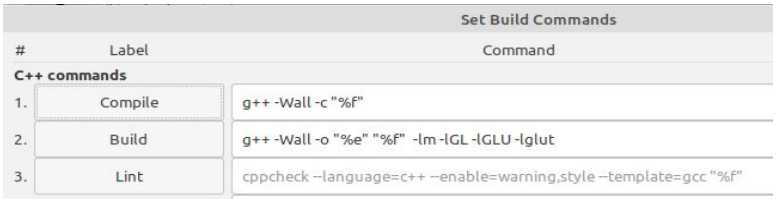
g++ -o cli148 cli148.cpp -lGL -lGLU -lglut

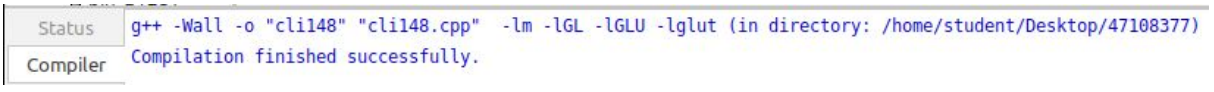
./cli148

Method2:

Open Geany, set the build command to **g++ -Wall -o "%e" "%f" -lm -lGL -lGLU -lglut**, and set the compile command to **g++ -Wall -c "%f"**.

Click the Build the current file button, and then click Run or view the current file.





References

Files from labs: loadTGA.h back.tga bottom.tga front.tga left.tga right.tga top.tga
Textures from Google Images: ufo.tga walls.tga
Tool for sketching mathematical equations: https://www.desmos.com/calculator
Physical equation: https://en.wikipedia.org/wiki/Solar_System