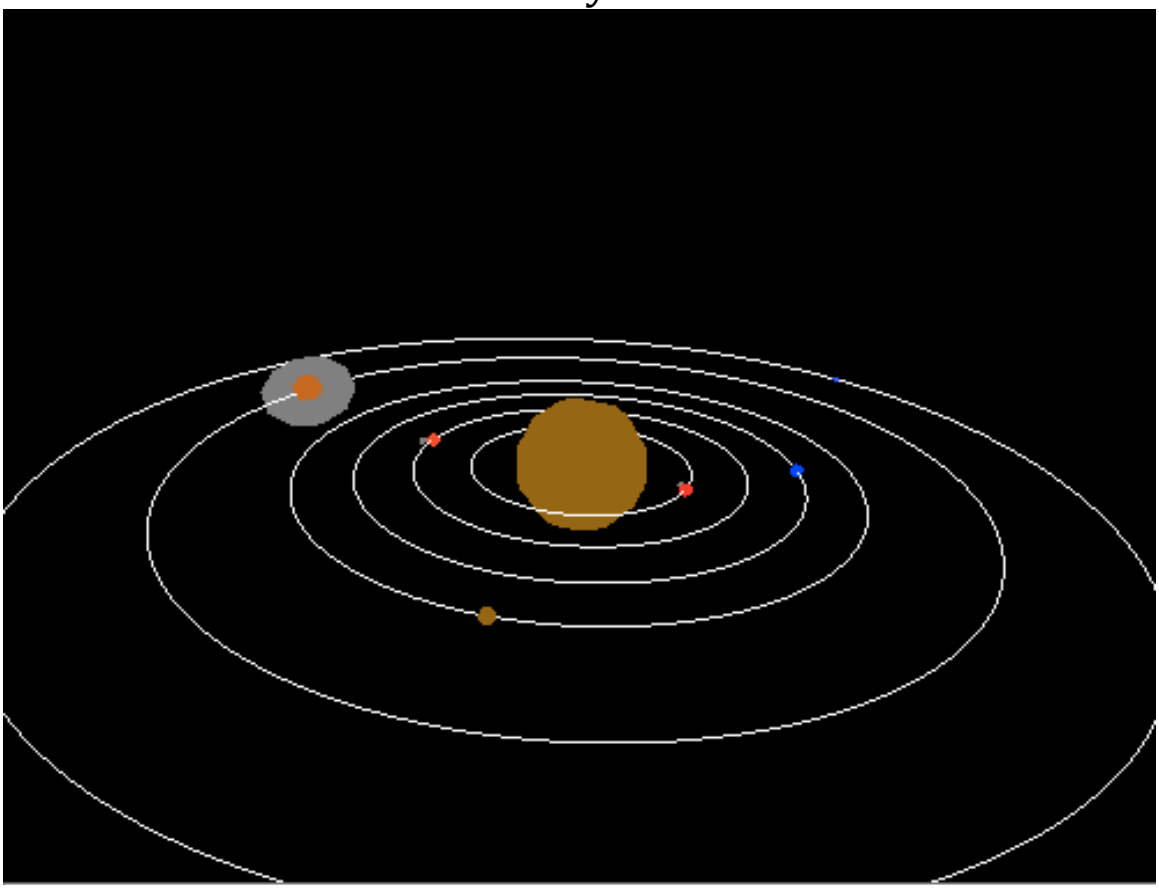
Solar System



Description:

You now are familiar with transformations and the order of operations that must take place. You now must understand how to build a hierarchy of objects that can inherit properties for your scenegraph (coming in a future homework – Assignment 3). In this lab, we will be building a solar system and understand how transformations are carried from one object to another through the modelview stack.

Your Task:

- You will build a sun, a few planets, and a few moons (or rings) that orbit the planets.
  - You will do this in the render() function in the solarSystem class.
- Understand how to push and pop onto the matrix.

Files Given:

main.cpp – You do not need to modify this solarSystem.cpp and .h – You will write the render function for the solar system.

OpenGL Commands Refresher:

glPushMatrix/glPopMatrix

```
1. glPushMatrix();//These can be nested
2. glPopMatrix();
3. glTranslatef(0,0,0);
4. glRotatef(0,0,0);
5. glScalef(0,0,0);
6. glMultMatrixf(const GLfloat
   *matrix)
7. glutSolidSphere(radius,slices,stacks);
```

Going Further:
Did you enjoy this in class assignment?

- Try adding alpha blending to the planets rings. Start looking into textures and other materials that can make the planets appear more interesting.
- Add satellites that can orbit the planets
- Add asteroids that orbit the solar system
- Create multiple solar systems that all rotate around a galaxy
- Add some interesting simulation
    - If a moon gets too close to a planet, will it get sucked into another planets gravitational pull and rotate about it?
- Add more planets with irregular orbits
    - Pluto for example has a much more egg shaped orbit