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 scene graph. In this lab, we will be building a solar system to understand how transformations are carried from one object to another through the model view 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.
  - Make your own crazy solar system! Physical laws do not have to apply to your imaginary universe!
- Become more familiar with the rotate, translate, and scale operations
- Understand how to push and pop onto the matrix.

Files Given:

main.cpp – You do not need to modify this  
MyGLCanvas.cpp and .h – This is the same code as the previous labs, but separated out into separate files for easier management  
SolarSystem.cpp and .h – You will write the render function for the solar system.
OpenGL Commands Refresher:

```c
void glPushMatrix(); // This can be nested
void glPopMatrix();
void glTranslatef(GLfloat x, GLfloat y, GLfloat z);
void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z);
void glScalef(GLfloat x, GLfloat y, GLfloat z);
void glMultMatrixf(const GLfloat *matrix);
void glutSolidSphere(GLfloat radius, GLint slices, GLint 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