**CSCE A385 Programming Assignment 4**

Computer Graphics

Shanan Almario

**Program Notes**

IDE: Visual Studios 2015

**Problem 1 Code Description**

The code outputs three objects and 3 lights. The objects are a torus, sphere, and ellipsoid, and the lights are a stationary light, orbiting light, and a moving spotlight attached to the camera.

**Interaction:**

*Keyboard*:

x: will move the space counter clockwise on the x-axis

X: will move the space clockwise on the x-axis

c: will move the space counter clockwise on the y-axis

C: will move the space clockwise on the y-axis

z: will move the space counter clockwise on the z-axis

Z: will move the space clockwise on the z-axis

*Spec Key:*

Up: move closer to objects

Down: move away from objects

Right: move right

Left: move left

**Problem**

The assignment was to draw a golden sphere, a ruby ellipsoid, and a silver torus under three lighting scenarios.

This assignment was structured similar to the cannon assignment where each object that is not a copy of another is a sibling in the tree. Those that are a copy of an object are children. All siblings have their own draw and build methods.

The lights were called in the draw methods. Each object had their own light object placed at the same spot as the other objects for each of the three lights. The light variables were organized in two structs, lightData and materialData. The lightData struct held the light color data and light position data, while the materialData only held the material and shininess data. Each of the different light objects have their own computation and initialization functions, and each object have their own initialization functions for the lights. The objects shared the computation function for their lights.

The orbiting light was created by referencing the example code from the shaded sphere. The stationary light was created by using the same structure as the orbiting light, but the light\_position does not change. The moving spotlight attached to the camera references the example code given in the phong torus code, and follows its example in the shader. Its position is equal to the eye view. There is a small problem with this light because the specular light can be seen, but the ambient light doesn’t seem to follow the specular light. This is probably because there isn’t a calculation for the ambient light to change.

The resize routine will change the image of the scene so that it matches the size of the window.

*Scene Structure Tree*