Final Project: 3D Scene in OpenGL

Vincent Snow

Southern New Hampshire University Computer Science  
CS 330 – Comp Graphics & Visualization  
Brian Battersby

June 18, 2023

**Development Choices**



The objects I selected for this scene were all fairly simple objects that could be created with primitive shapes, two of which would only require one shape. The two complex objects were challenging but fun to create and rotate properly. The crystal pyramid and candle were created with one shape. The palo santo bundle is made of two cubes (rectangles) for simplified sticks, and a torus with a cylinder through it as the twine wrapped around them. The dousing bead was the most complex, made of two spheres, three cylinders, a torus, and a cone as well as three different textures. I wanted to try using many different shapes to see how they transformed and look at how they were all created by drawing triangles. My lights are represented by torii because they match the shape of the lights in the room where the photo was taken.

**Navigation**

The camera in the scene uses a common kind of right-handed keyboard control scheme seen in games and 3D interactive environments where the AWSD keys act as movement arrows (on the x and z axes), and the mouse controls the pitch and rotates the camera. The scroll wheel changes the speed of the camera, and the Q and E keys move the camera up and down on the y-axis. There O and P keys change the display from an orthographic 2D projection to a 3D one. This allows for a full range of movement around the scene and views from different angles for all of the objects at whatever speed desired.

**Custom Functions**

This code makes use of custom functions for many parts of it. From the creation of objects and textures to the movement of the camera, these functions cut down on repetition and shorten the code throughout. One function can be used repeatedly to load a shape model matrix and then apply transformations or textures to it, and another can be used repeatedly to map a texture to a shape and apply it for drawing. Camera controls are done through a header that processes the input of certain keys and then passes that information back to render a frame. Another function performs the math necessary to calculate the color under Phong lighting and can be called to apply new lights. I have reused the headers and custom functions for various assignments and could reuse them all for creating a completely different scene just by copying and pasting or including a file. The comments in the code make it easy to search or find blocks that apply to different elements of the image and can be easily re-used or altered as a template.