3D Scene Justificaiton:

I picked common objects around my house to try to create a scene. I picked a combination of shapes to try to challenge myself. For the book I just used a rectangle with no bottom as we wouldn’t ever see the bottom. I used a sphere for the ball and a cylinder for the candle. For the cords my initial idea was to use a torus based on some work I had don’t in GLUT previously I didn’t think it would be to big of an ask. After realizing that all the shapes need to be added programmatically I opted to use two cylinders and linked them together.

Explain how use can navigate:

The mouse dictate your look at direction and the WASD keys move you forward, backwards, left and right. The Q and E key control vertical movement. The mouse scroll wheel controls the speed of movement and pressing ‘P’ toggle between perspective and orthogonal views.

Explain the custom functions:

I have a few custom functions in the code. I will touch on the cable function. Essentially it contains coordinates for two cylinders connected together on the top. I used it to create a loop of cables on the upper right hand corner of the book. Its reusable in that I could have easily used this for the candle body as well but decided not too as the geometry would just be hidden.

Issues:

I spent a good deal of time researching and comping up with ways to get the vert locations for this project with limited success. I ended up utilizing some of the algorithms at <https://www.songho.ca/opengl/index.html>. However, after storing the verts, normals and texcoords in a std::vector<float> and trying std:vector<GLfloat> I found I was unable to use the vectors like I had previously used the GLfloat[]. Instead I used a temporary function to print out the vertices normal and texcoord array to the console and pasted them into the cylinder and sphere verts arrays.