The scene that I was recreating was an office space. The first thing I did was make sure the floor and walls were in place as these would provide the backdrop for the scene. The desk in the room is created using a cube mesh with 4 cylinders that connect with the floor plane. The legs have a metal texture to try and match the scene that I was recreating. On the desk are three objects. On the left is a plant pot. This is created from a cylinder mesh for the base with a sphere mesh that is on top of it. Another cylinder is then placed overtop of the sphere to give it a container appearance. A textured plane makes the soil that several cylinders rise out of. The leaves are two prisms that link up to give a leaf shape. The computer is made up of two rectangles and a plane that is textured to display what mode the user is currently in. I also added lighting to the screen based on what mode the user is in and the color of the texture. The pencil cup is another group of 4 cylinders. The cylinders or pencils inside are scaled down and rotated as they are seen in the reference photo. Finally, there are two white pyramid lights that are on the wall.

The user can navigate the scene using the keyboard and mouse. If the user wants to move around, they use W for forward, S for backward, D for right, and A for left. If you want to move up or down, you will use Q or E respectively. The program also uses the mouse to change the speed that the user moves around the scene. The mouse movements are also used to allow the user to look around where they are in the scene. There is also functionality for the user to switch between a perspective mode using the P key which will allow the user to “walk around”. They can leave that mode and return to Ortho mode by pressing the O key.

This program uses a class to handle object creation. It is called meshes and it automates the process of plotting out points for different meshes. When I first started the OpenGL project, I was mapping out each point and had the program draw it all at once. This took A LOT of time. The meshes class handling this for me meant I just needed to translate the image to where I wanted it on the scene. The other class is the camera class. This kept the code organized for handling camera controls. Both classes can be easily implemented into any OpenGL project.