

This project is a scene of my desk. Each of the objects I included is something I am very familiar with. They are things I see and use nearly every day. I had some extra objects in my original photo of my scene that I did not include in my final scene. I did this on purpose, as I knew that as I learned more about what we were doing in the class I would learn what was achievable and could remove as necessary. The largest example of this is the plasma ball I originally had in my scene. I had this in up until it was time to add textures; I then realized that it was far more complicated to make a transparent, glass-like texture than I had originally guessed it might be. After looking at some resources on the topic, it requires rendering the depth maps on a per-object basis, making custom shaders for the glass, and a lot more.

Another thing I would have liked to implement is shadows, as one of the things that bugs me most about the scene is that light kind of hits everything instead of stopping when it should. After looking at LearnOpenGL's tutorials on shadows, I quickly decided that it was a project for another day. It takes quite a bit of time and effort to create shadows.

I believe that navigating my scene is intuitive for those who have experience moving around in 3D environments. I also followed the instructions given for what the control recommendations were. These include, moving forward, backward, left, and right using the W, S, A, and D keys respectively. Moving up and down in the scene can be done with the Q and E keys. Looking around the scene is done with the mouse. As the mouse is moved around the screen, the camera follows. To prevent issues like ending up upside-down, there is an 89-degree cutoff for looking straight up and down. Without this, as you move your mouse forward, the

camera would move past the directly vertical look, and start looking behind you, but upside-down.

While in a perspective view, scrolling forward on the mouse wheel increases the movement speed of the camera, while scrolling down slows down the movement speed. In an orthographic view scrolling zooms in and out from the scene. Scrolling up zooms in and scrolling down zooms out.

A click of the left mouse button will reset the view to perspective and place you back at the camera's origin from when first starting the program.

The last thing that I implemented is switching between a perspective and an orthographic view. This can be done with the P key.

Each of the texture files are grabbed and created from the main function. I used 12 textures. I had so many it was getting tedious to keep grabbing the same few lines for each texture as well as taking up a lot of unnecessary space. I created an importTexture function. It looks like this:

```
void importTexture(const char* filename, GLuint& texName) {  
    const char* texFilename = filename;  
    if (!UCreateTexture(filename, texName)) {  
        std::cout << "Failed to load texture " << filename << std::endl;  
    }  
}
```

Importing a texture file looked like this before:

```
const char* texFilename = "resources/textures/WoodFloor007_2k-  
PNG/WoodFloor007_2K-PNG_Color.png";  
if (!UCreateTexture(texFilename, gTextureId_LightWood)) {  
    std::cout << "Failed to load texture " << filename << std::endl;  
}
```

Now it looks something like this by calling the importTexture function:

```
importTexture("resources/textures/WoodFloor007_2k-PNG/WoodFloor007_2K-  
PNG_Color.png", gTextureId_LightWood);
```

This allowed it to be on a single line and made it far less confusing to bring in a texture.