The final project for this course was a culmination of all the different graphics rendering techniques that we utilized and implemented over the course of this class. The assignment requirements were to create a 3D scene of our choice, and render objects with textures, lighting, and color to represent the chosen scene. The scene I had chosen to replicate is my computer office setup. The scene consists of my desk, keyboard, mouse, monitor, and a cup with pens stacked inside. This scene was selected to simulate an everyday scene and a familiar environment.

Starting with the desk, this serves as the foundation of our scene. We used the plane object to serve as the desks model. After we implemented the plane object, we then focused on mapping its coordinates to the scene and beginning to apply textures. I found images of wood grain and plastic texture while searching for textures that I could use in my scene. I mainly wanted to go for a wooden texture that really simulated a realistic wooden desk. After a few attempts, I implemented an image that covered the object nicely and added a touch of realism.

Moving forward, we also focused on making sure the object that was created next sat on top of the object nice and flush, really simulating the computer desk scene and adding a sense of realism. I decided to design a keyboard to lay on the desk. The foundation for the keyboard was a square that I applied texture to and mapped it to see directly on top and centered on the desk. For the keys, we had to map them to be on top of the keyboard, and I implemented code to lay the keys spread out evenly on top of the keyboard, with smaller cubes to represent them. So, I had a giant cube for a base with other miniature cubes on top for the keys.

The monitor and mouse I will lump together, because these implementations were very similar. These three, while they look unique, are all boxed meshed laid out on one another to represent a monitor, and a miniature box on our mouse to represent the mouse wheel. These implementations were similar in mapping, as we had to lay the objects flush on the desk. The biggest standout here is the coloring I used, the blue screen really sticks out and makes the scene standout.

The more complex implementation was the cylinder cup with cylinder pens inside. The cup was added on the left side of the desk, and I wanted to add a bunch of different looking pens inside. So, what I did was I made a collection for big and small pens, and then implemented a loop to place the different pens from different collections inside of the cup.

To meet the project requirements, I had to focus on implementing the major functionalities that were required by the project. Camera navigation was implanted to give users control over their view of the 3d scene. The WASD keys were assigned basic control over the camera, but the Q and E keys had to have vertical functionality, so I had to manually implement code to trigger when these keys were pressed during the viewing session. Having manual control of not only the camera, but the verticality of the viewpoint really adds to the depth and a sense of exploration to the scene. Thanks to this implementation, users can navigate the scene with ease.

Lighting was a critical component to the scene. By adding different forms of lighting, like point lighting, we can illuminate our objects and colors, and create a visually compelling scene. The camera and lighting work hand in hand, so it’s important to choose the right lighting for the right situation. For example, your camera could potentially block a pinpointed light at the right angle, so implementing these sources carefully is critical. I had to implement my own methods to create the lighting, using the milestone assignment as a guideline to implement the light source. Here is an example of some of the code I used to create my own lighting method:

void SceneManager::SetLightProperties()

{

    //m\_pShaderManager->setBoolValue(g\_UseLightingName, true);

    m\_pShaderManager->setBoolValue("pointLights[0].bActive", true);

    m\_pShaderManager->setBoolValue("pointLights[1].bActive", true);

    // Set light 0 properties

    m\_pShaderManager->setVec3Value("pointLights[0].position", 3.0f, 14.0f, 0.0f);

    m\_pShaderManager->setVec3Value("pointLights[0].ambient", 0.01f, 0.01f, 0.01f);

    m\_pShaderManager->setVec3Value("pointLights[0].diffuse", 0.4f, 0.4f, 0.4f);

    m\_pShaderManager->setVec3Value("pointLights[0].specular", 0.0f, 0.0f, 0.0f);

    // Set light 1 properties

    m\_pShaderManager->setVec3Value("pointLights[1].position", -3.0f, 14.0f, 0.0f);

    m\_pShaderManager->setVec3Value("pointLights[1].ambient", 0.01f, 0.01f, 0.01f);

    m\_pShaderManager->setVec3Value("pointLights[1].diffuse", 0.4f, 0.4f, 0.4f);

    m\_pShaderManager->setVec3Value("pointLights[1].specular", 0.0f, 0.0f, 0.0f);

}

To keep the codebase clean and modular, I added the methods I created in their proper locations and files. For example, when I implemented code for the creation of shapes, I did so in the appropriate location where the other shapes had been initialized as well. The codebase provided already had some structure, so adding to it was relatively straightforward as everything had already basically been architecturally mapped out in the codebase. Obviously, objects and methods that had been added to the code were done so in the appropriate folders, as some code can be interdependent on code that resides elsewhere, so rigorous testing and documentation is necessary to keep track of the changes I made when developing the scene.

In conclusion, my scene was designed to replicate an everyday scene of an office work desk. I had implemented objects, lighting, textures, and coloring to the scene to add depth and immersion, representing the scene with realism. This process was a mix of simple implementations as well as more complex details. The camera not only provides basic movement but verticality as well. By carefully implementing these features and focusing on project requirements, we created a vibrant 3D scene that users can navigate with ease and have an immersive experience.