Dalton Rose

CS-330

2024/12/14

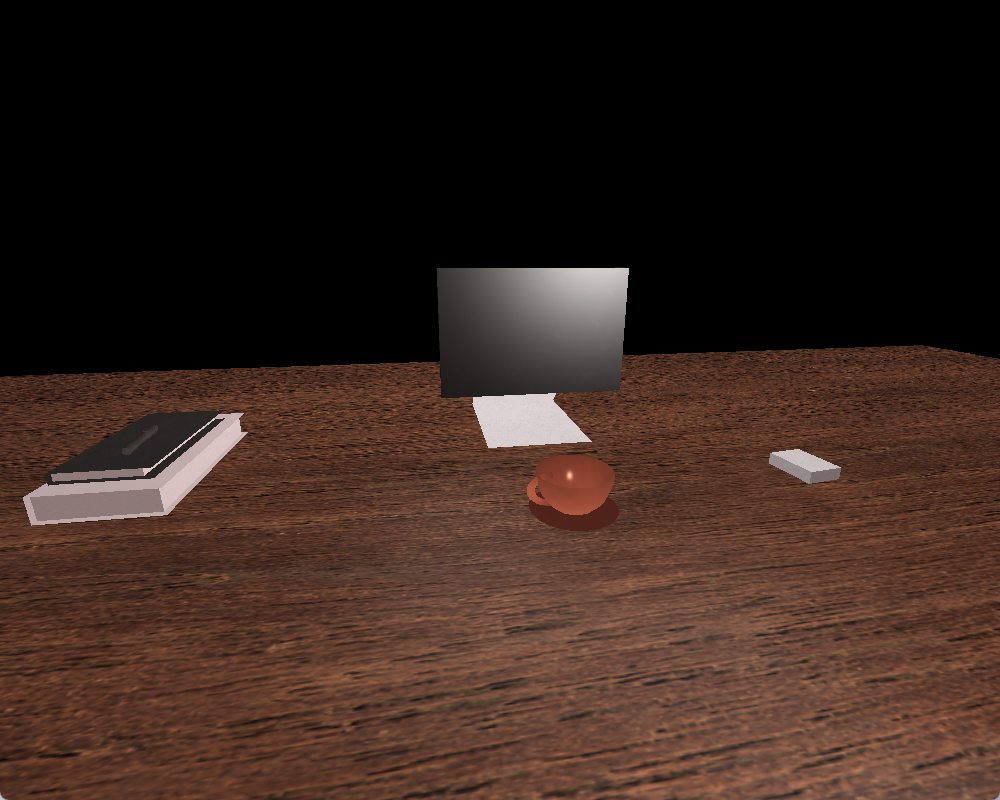
Final Project Reflection

The 3D Scene for my final project depicts a simple office desk setup, with a computer monitor, mouse, stack of notebooks, a pen, and a cup. When deciding which objects I wanted to adapt from my reference photo, I wanted to choose a mix of simplistic objects and complex objects. However, my initial ideas were too simplified and almost every simple object transformed into a complex object with time. For example, I had originally planned to use a single cylinder for the pen. However, I ended up using two differently sized cylinders to model the pen with the cap as an additional detail point.

I have also taken some liberties with texturing and lighting the objects to create a more interesting scene. For example, my reference photo was very monotone, and I corrected this by using a wood texture for the desk and an orange texture to the cup to add a splash of color. Additionally, the lighting in the reference photo was very bright and sterile. I decided to tone down the brightness of the scene and add a slight warm tone to some of the lighting to create a different atmosphere. With these changes to the texturing and lighting, my 3D Scene still resembles the reference photo but takes on a different feel.

Users can navigate my 3D Scene by using the WASD keys on the keyboard to move around. Users can also use the mouse to move the camera view. By using the Q and E keys on the keyboard, users can raise and lower the camera respectively. Users can adjust the speed of the camera movement using the mouse scroll wheel. These input methods allow users to effectively navigate through the 3D scene.

The Project template was a very helpful tool when structuring and completing this project. The functionality of the project was already separated into several different files which improved readability and reusability. For example, the changes I made to create the scene were primarily to the SceneManager.cpp file which contains information about the scene such as objects and their properties, textures, lights, and more. In the ViewManager.cpp files, some changes were made to add additional camera functionality like vertical movement and speed changes. By dividing the project like this, changes to the camera movement could still be applied even if you were to create an entirely new scene.

However, there are ways in which I could have improved the organization and reusability of my code. All shapes for my entire scene is contained within the single RenderScene() function. While there are comments to explain each shape, it is a long function that may be difficult to understand at first. If instead I had broken up each object into its own function like RenderBooks() or RenderCup(), it would have improved the readability and reusability of my scene.