Dylan Coulter June 22, 2024

ID#1896062

CS330

**CS330 Final Project Reflection**

The scene that was chosen to be recreated in 3D using C++ and OpenGL programing contained four different objects. The objects that were chosen were a box of cereal, a spoon, a bowl, and a plane for the other object to rest on. The objects were a good choice as they would be challenging to recreate, but achievable. The spoon and the bowl would need more than one object to be used to accurately recreate them. To start the project a plane was created, correctly placed in the scene, and colored brown. Next, the bowl would be create containing three different 3D objects a half sphere, cylinder, and a torus. The bowl would be colored white for the time being as the other objects were created. After this the cereal box was created and placed in the correct position and colored red. Lastly the spoon was created which was the most difficult object to create. The spoon was made up of a half torus, half sphere, and a cylinder. Once these objects were scaled, rotated, and placed in the correct locations in the scene you could see how it was starting to resemble the original image, but was lacking something. This something was texture and material shaders on the differing objects in the scene to help us create an even more accurate representation. For the plane in the scene a texture like wooden table was resourced with a shader material like wood applied. For the bowl and the spoon, it took a while to get resource the textures that would give them the most realistic appearance in the scene. In the image we chose both the spoon and bowl have the most shininess and reflection of light, so it was important to include this in the material shader for both. To create the texture for the cereal box I took a photo on my phone of the box used in the original scene. Then I converted it to JPG added it to the texture file in my project and sized it to 1024 by 1024 pixels. The material shader for the cereals was developed with the idea of how shiny the waxy coating on a cereal box is. After all this it was time to add the final piece to help make our scene as accurate as possible and this is the lighting. I decided to go with three different light sources two point and one directional. Almost all the light in the scene is coming from the upper right which is how were place our three lights. Our light sources are in various position in the upper right of our scene to help create the same effect as the original image. The direction and the use of the three different types of light sources ambient, diffuse, specular, and their color with each light help us obtain the most accurate representation of the original scene.

The creation of our scene up until this point was accurate to the original image, but we were only able to view the scene from the camera position preset. We would now add camera movement to the scene to allow the user to navigate our work. First, we would add movement that can be done with the mouse. This functionality would work when the user would have the mouse in the display window. The program would track the movement and the user would navigate the scene. We also added movement with the W, A, S, D keys which the user would press to move the camera forward, backward, left, and right in our scene. Four other keys were added to the program to help the user navigate. The Q, E keys we added so the user could move the camera up or down within the scene. The O, P keys were added so the user could quickly switch between perspective and orthographic view with the press of a button. To help with the camera movement in the scene we create a way the user can decide how fast the camera speed will be when using these controls. To do this we added functionality to the scroll wheel on the mouse which the user can use to adjust sensitivity. While coding this project a focus was the reusability of the code that was being worked on. When looking over the code in both scenemanager.cpp and viewmanager.cpp you will see how everything we spoke about is kept together in the code making adding or deleting simple. For example, in the viewmanager.cpp class we have all the camera movement code and with this all the key functionality code is together. This makes it easy to just add a new key with some sort of new functionality. We could add keys that quickly change the angle of the camera in the scene, and this could give us the ability to add multiple new angles to allow the user to navigate even quicker. The same goes for scenemanager.cpp which contains all the objects, textures, materials, and lighting. All of the lighting code is kept together and commented appropriately making the code modular and this also goes for the textures, materials, objects, keyboard events, mouse events, and scroll events. Towards the end of the project this structure made it easy to fine tune any areas of the scene that needed it. One of the last things I did was add a new plane to the scene to represent the backdrop in the original image. It was quick and easy for me to create, position, texture, material, light this object which help give an even more accurate representation.

**Original Image**

A bowl and spoon on a table

Description automatically generated

**Recreated Scene**

A bowl and spoon on a table

Description automatically generated