Reference Image:

A group of salt and pepper shakers on a table

Description automatically generated

Lighting:A screenshot of a video game

Description automatically generated

Final Scene without Debug Lights:

A group of objects on a surface

Description automatically generated

Final Scene in Wire Mesh View:

A group of objects on a black background

Description automatically generated

***My scene consisted of a number of unique and complex objects that entailed a number of design choices:***

**The Table** was created using a plane. The reason for this was because the scene is intended as a showcase, and we weren’t building out the entire room. If the intent was for a more exploratory and complete room, then the table would’ve been built out of a cube scaled to a rectangle. The plane required a single texture mapped to the surface that resembled the wood grain of the reference table.

**The Shakers** for salt and pepper were created using Cylinders, the cylinder for the body of the shakers was in the form of an octagon and the caps were in the form of a circular cylinder. The design decision behind this was to make the shakers representative of reality but also grant a chance to explore the different algorithmic generations of cylinders. Creating the functions for these shapes was challenging but rewarding. Especially as it relates to generative texture mapping.

**The Salt Container** was an extension of the Shaker Body objects. Since the Salt container was a cylinder, it required more mapping points than the 8 points that the Octagonal Cylinders of the Shaker Bodies required. The Texture Mapping for this object was quite challenging since the Logo for the Salt required legibility and being stretched. I ultimately decided to scale the texture by a factor of two to make it easier to read.

**The Pepper Container** was represented by a cube mesh scaled to the shape of a rectangle. The main learning curve for this object was figuring out how to remove the texture (sample 0.0f for u and v coordinates) for the proper faces. I only required the logo to be on the front and back face. Additionally, I had to reverse the mapping coordinates for the back face of the Pepper shaker so it could be read properly.

**The Two Textured Spheres and the Sphere Holder** were additional objects I added. While I was coding the meshes for the other objects it occurred to me, I had not made a mesh for a sphere and desired to. Through a lot of trial and error I was able to generate them. It gave me an opportunity to be creative and add some interesting textures to the two different spheres. Additionally, while creating the sphere holder I had a chance to use the original table\_top mesh I had sourced for the table. The gray tabletop mesh did not match the reference image, so I did not use it for the plane. Recycling it into the sphere holder object was a fun way to use what I thought to be a handsome wood grain texture.

**Explain how a user can navigate your 3D scene**. As you compose your thoughts, discuss how you set up to control the virtual camera for your 3D scene using different input devices.

***My Scene provides a number of interesting ways to interact with the environment:***

The Scene that I created allows several ways to navigate and fully appreciate it from every angle. Utilizing the W, A , S D keys you can move forward, backward, left and right. Utilizing the Q and E keys you can move up and down in the scene. The Mouse Cursor allows for looking in all four directions. The Mouse Scroll Wheel allows for toggling the speed of the movement, this is very advantageous if one is feeling impatient while exploring the scene.

Lastly, for sake of curiosity more so than utility two other functionalities were incorporated. Using the TAB key, you can toggle the wireframe view on and off. I found this to be a really fascinating piece of the coursework this term and was happy to incorporate it. Viewing the wireframe allows one to really appreciate the skeletons of these objects and how much coordination has to occur between the vertices and indices mappings. In addition, the P key allows for toggling orthogonal and perspective views, this is an additional curiosity that allows the viewer to explore how much the automatic calculation of perspective plays a role in computer graphics.

***Behind the scenes, many design decisions occurred to modularize, abstract and enhance the code organizationally to make future development more robust:***

During week 6 of the semester, I took the time to fully organize and modularize the code. Initially the main body of code consisted of the Shaders, Mesh Functions and Texture Loading. This resulted in a very cumbersome codebase that was hard to navigate.

**To Modularize the Shader Process,** I put all of the Shader Text into a separate folder in .glsl files and loaded them into the program indirectly instead of in the main body of code.

**To Enhance the Mesh process,** I created a Mesh Header and Mesh\_Utility file that allowed Mesh Creation to occur in a different file. This removed a large chunk of code from the main file.

**To enhance Texture Loading** I relocated the Texture Loading function to the Mesh\_Utility file and stored the textures in a separate folder.

Through these enhancements to the code base, I was able to greatly improve my developer experience with the project.