Reflection

When creating my project’s scene, I chose to include objects that have great significance to me. I play drums in a group with my friends, so I use a practice pad, drumsticks, my earbuds, and my cellphone to keep up with my timing and rhythmic patterns. The practice pad I use is a decagonal prism for the base and rubber face. The earbud case has an off-center oval where the case opens that I wanted to include to improve the visual, so I created a custom texture with a sine curve along the top. All of the textures I used were covered within the creative-commons license, but they didn’t exactly fit the aesthetic I wanted in my scene. Because of this, I implemented quite a bit of blurring and color adjustment to each texture. The wood grain needed the most adjustment, as it did not look life-like enough for the scene.

A user can navigate around my scene using the W and S keys to move their position forward and backward, A and D keys to move left and right, and Q and E to move upward and downward respectively. These movements are all relative to the current orientation of the camera. I also included inputs to change between Orthographic and Projection views using the O and P keys respectively. When using the Orthographic view, I locked the camera orientation forward-facing horizontally to the plane, and I eliminated the ability to adjust the camera orientation while still allowing upward, downward, leftward, and rightward movements. The Projection view allows full control of the camera’s position and orientation. The user can adjust the camera’s orientation using mouse movement, and the speed of positional movement can be adjusted using the mouse’s scroll wheel.

A custom function I used within my program was the implementation of a decagonal prism mesh. I created a decagonal prism mesh to properly apply texture and lighting normals while remaining true to the shape of the practice pad. This is now reusable when creating any decagonal prism meshes, as all the texture coordinates, normal values, and positional values are stored within the method. To maintain code organization, I used appropriate commenting to explain each segment of the code and its purpose. The code was modularized using texture, lighting, and object material creation methods that could be reused throughout the code as needed.