**Final Project:**

**Reflection**

Taylor R. Jones

CS330

SNHU

Prof. Brian Battersby

4/20/2025

**Introduction**

My 3D scene is inspired by everyday objects found around my home, chosen for their variety in texture, shape, and interaction with light. The goal was to create a realistic and visually engaging representation by accurately capturing proportions and positioning, applying custom high-resolution textures generated with AI, and designing a lighting setup to simulate morning sunlight streaming through a window. I really enjoyed working on this project and dedicated extra time to writing modular, organized code to streamline the final scene setup.

A bottle of soy sauce and a ball

AI-generated content may be incorrect.

Figure 1. Early staring point of the scene versus reference image.

**Development Choices**

The objects I selected for my scene include a soy sauce bottle, a metal funnel, a tennis ball dog toy, and a stress ball shaped like a stick of butter. I knew I wanted four objects, with at least two being more complex, so I started with the funnel since it could be effectively created using the available mesh types. The soy sauce bottle was an appealing choice because it allowed me to experiment with transparency to simulate glass, and I appreciated the opportunity to add small, detailed touches that make the object stand out. I added the tennis ball and butter stick to introduce more shape variety and to explore how different materials interact with light. Since the bottle and funnel both have smooth, reflective surfaces, I chose the matte finish of the ball and butter to create a more balanced and visually interesting scene.

Then, I built out each object with basic meshes and fine-tuned object proportions and positioning. Once everything was in place, I began applying simple textures as placeholders so that I could work on the scene lighting. I implemented a directional light from slightly above and to the right of the scene to simulate the morning sun shining through my front door which was the light source for my scene reference pictures. From there, three point lights were strategically placed to introduce ambient, diffuse, and specular light into the scene from the right side. I also implemented a spotlight with various intensities, but I didn’t like the resulting flashlight effect and removed it to preserve the morning lighting in the scene. One trick I found helpful was utilizing small temporary sphere meshes as debugging objects to help with dialing in precise vector coordinates for light placement as seen below:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Figure 2. Advanced lighting being added to the scene.

Once I was happy with the lighting in the scene, I began generating custom high-resolution seamless textures with ChatGPT to make the scene as realistic and close to the reference image as possible. I went through several dozen textures to get the finished product and utilized advanced texturing techniques to bind textures to separate faces on objects like the stick of butter and bottle cap to improve fidelity to the scene and add interesting visual details such as the ridges on the cap or the folds on the ends of the butter.

A yellow box with blue text on it next to a green ball

AI-generated content may be incorrect.A red cylinder on a wood surface

AI-generated content may be incorrect.

Figure 3. Complex texturing techniques

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 4. Complex texturing code snippet allowing textures on specific object faces

**Navigating the Scene**

Users can navigate the scene using a mouse and keyboard. The camera can be toggled between Perspective (P) and Orthographic (O) mode with their respective key bindings. For example, when a user presses O, a Boolean value is set to true to enable orthographic mode, and the camera is moved to a viewing angle from the right of the scene even with the plane the objects sit upon. There is also Orthographic side (L) and top (T) view implementation. W and S allow you to move the camera toward or backwards from the scene. Q and E allow users to pan up and down. A and D allow panning from side to side. The mouse can be used to look around the scene and the scroll wheel controls movement speed.

A computer screen with white text

AI-generated content may be incorrect.

Figure 5. Camera perspective controls

**Custom Functions**

Modularity and reusability were prioritized from the earliest stages of the project. Functions like RenderFunnel() and RenderBottle() were designed to support easy translation, making each object straightforward to reposition within the scene. A key focus throughout development was realism, and one of my first steps was creating a flexible system for object positioning. For example, I introduced a funnelOffset variable defined as glm::vec3(4.7f, 0.0f, 0.0f);. By adding this offset to each component of the funnel, I was able to move the entire object by adjusting a single value. This same technique was applied to other scene elements, enabling modular and efficient transformations across the project. Additionally, I configured texture materials to be reusable, eliminating the need to define unique materials for every object. With SetShaderMaterial(), I could easily assign glossy, matte, or medium-shininess finishes depending on the desired surface appearance.

**Final Comparison:**

**A screenshot of a computer

AI-generated content may be incorrect.**

