Scott Cain

4/23/3035

CS 330 Final Reflection

**Reflection Document: CS 330 Final Project**

The CS 330 final project challenged me to develop a 3D scene using C++ and OpenGL, replicating a 2D image that I chose. I demonstrate my ability to generate accurate representations of 3D objects, apply textures, design lighting, enable interactive navigation, and follow best practices in code development.

**Object Selection**:  
To create my 3D scene, I selected objects that closely resembled 2D. These objects mostly had simplicity for low-polygon modeling. Objects were constructed using basic shapes such as approximate the features. I did have issues making the bowl, I did make have a longer y axis to give more depth.

**Modeling Approach**:  
I prioritized organized geometry and optimized polygon spacing to ensure that all 3D models adhered to the project's guidelines.

**Texture Application**:  
Textures were selected based on their resolution (1024 by 1024 pixels or higher) and royalty-free availability. I applied butter, potato skin, stainless steel, and plastic.

**Lighting Design**:  
To create a polished visualization, I designed two light sources: a white directional light and a colored point light. These light sources highlighted the scene's features and improved the user’s ability to explore the objects. The Phong shading model was utilized to balance ambient, diffuse, and specular components. All the objects were well lit from multiple angles.

**Explain User Navigation**

**Input Controls**:  
The navigation system allows users to explore the 3D scene interactively. Using WASD keys, users can move forward, backward, left, and right, while QE keys control upward and downward motion. The mouse cursor enables changes in the camera's orientation, allowing users to look up, down, and side-to-the-side.

**Camera Dynamics**:  
Nuanced controls were implemented for smooth navigation, including adjustments to pitch, yaw, and roll. The mouse scroll feature lets users modify the speed of movement, enhancing their exploration of the scene. Additionally, I implemented functionality to switch between orthographic (2D) and perspective (3D) views, providing flexibility in visualizing the scene.

**Detail Custom Functions**

**Functionality Breakdown**:  
Several custom functions were developed to make the code modular and reusable. For instance, functions for object placement automated the positioning of 3D models based on X, Y, and Z coordinates. Similarly, functions for applying textures ensure consistency across different objects. Input handling functions simplified the integration of user controls.

**Code Optimization**:  
Best practices were applied throughout the development process. Code formatting included proper indentation and spacing to maintain readability. Descriptive comments were added to provide clear explanations of each function’s purpose. Logical organization ensured the program ran efficiently, with tasks appropriately divided into modular functions.