**Module 7 - Final Project Reflection**

**Jonathan Wolanyk**

**CS-330: Computer Graphics and Visualization**

**Doctor Loay Alnaji**

**February 19, 2023**

**Justify development choices for your 3D scene:**

While developing my scene, I made a few development choices that allowed for the project to be completed promptly and in a manner that complies with the activity’s guidelines. The rubric requirements were addressed sequentially to ensure that the final result was of high quality from a code and aesthetic perspective.

The first requirement was to create low-polygon 3D representations of real-world objects. My scene was that of a chess set with various pieces. I utilized cylinders, planes, and spheres to create cohesive models for each piece. One development choice I made was to swap the rook in my reference image for a pawn since the embattlements of the rook would have taken quite a long time to recreate, as well as potentially pushing the model over the 1,000 triangle requirement outlined in the project’s instructions.

The rubric also requested that students accurately project textures to the 3D models. Since chess has two sides (black and white), I wanted to include pieces of each color to add more realism to the scene. As a result, I found high-quality texture images for wooden black and white pieces and projected the textures to the pieces appropriately. Additionally, I created and applied a chessboard texture to the plane underneath my pieces to represent the chessboard. Once the chessboard texture was applied, I positioned my pieces accordingly to simulate a chess match.

To apply lighting to the scene, I wanted to create a warm feel to the match since the wooden chess pieces and the board had created a sense of elegance already. As a result, I applied two point lights with warm yellow coloring to accentuate the wooden chess pieces and board.

**Explain how a user can navigate your 3D scene:**

Navigation in the scene is a crucial element of the project since one component of the beauty of a chess set is being able to view the game and pieces from multiple angles. To allow users to move throughout the scene, I applied horizontal movement by implementing WASD input keys to move forward, left, back, and right respectively. To allow users to move vertically, I allowed the input keys Q to ascend and E to descend.

Mouse input was vital in creating a smooth navigation experience as well. As a result, I created logic in the code that has a callback for the mouse position and scroll wheel to allow for changing the camera's orientation from mouse input, as well as changing the movement speed by having a callback for the mouse scroll wheel.

Another requirement of the project was to allow users to tap a key to change the viewport display of all objects in the scene between orthographic and perspective views at will. To satisfy this requirement, I allowed users to press the P key to change between the views. For future iterations of such projects, I want to ensure that when a user presses the key, the scene only accepts input once. Currently, the perspectives may flicker since the input is being checked for each frame while the input is checking whether the P key is down currently.

**Explain the custom functions in your program that you are using to make your code more modular and organized:**

To make my code more modular and organized, I implemented a few pieces of code as well as scripts and shaders. For instance, I found that I was making many cylinders for the chess pieces, so I utilized a cylinder script to make the process modularized. Instead of writing the logic to calculate the circumference, height, etc., of the cylinder, I can simply state the scale and location of the cylinder to place a new cylinder where I would like and at the correct scale for the object I am creating. I did the same for spheres in the scene as well.

In addition to modularized object creation, I implemented some variables to aid in building my scene faster as well. Since the objects in the scene remain at the same X and Z coordinates, I would make variables for each piece’s X and Z values. This made it much easier to position the pieces once I applied the chessboard texture since I had to move each piece to the center of a square to simulate the match in action.\

If I were to further modularize my code, I would want to develop functions for each kind of piece, such as a pawn, bishop, or queen. While I was able to use the same shapes and place them throughout the scene in a modular fashion, I would have enjoyed being able to simply request that a pawn be created at a given location and have the piece appear in the scene.