**Final Project Reflection**

Samuel A. Meade

Southern New Hampshire University

[CS-330-T2834 Comp Graphic and Visualization 22EW2](https://learn.snhu.edu/d2l/home/1196969)

Gene Rodriguez

12/07/2022

A picture containing text

Description automatically generated My 3D scene illustrates a low poli card game. Typical of most card games, a deck, notebook, and pencil are required to play and keep score. This collection of items can be represented in a handful of primitive shapes. The pencil, being the most complex, can be broken down into a cylinder and a pyramid. The deck of cards is a rectangular cube. The notebook could be represented with different interpretations, but I decided to use a rectangular cube base with a raised triangular prism to represent how the pages are elevated towards the spine of the book.

Throughout the milestones for the assignment, I broke each plane of each object apart, commenting their object, direction, and relation to move into future functionality more easily. This made texturing and lighting less of a hassle. Each vertex was separated into eight floats, the first three are the XYZ coordinates of the point, the second three are the XYZ coordinates of the lighting/positioning, and the final two floats are the repetition and positioning of the textures.

Utilizing mouse and WASD + QE camera movement, a user can mouse over the direction the camera should face and move through that space with the WASD keys. A user can then adjust proximity, or zoom, through mouse drag or QE keys. Combining these camera functions allows the user to seamlessly move around, over, or under the entire scene from any angle.

My code is oversaturated with vertices and logic inside the Source file. All of the textures and triangles are central to that file, but clearly differentiated by concise comments. In a previous milestone, I was playing around with modular objects, attempting to create reusable base classes and a class for each unique primitive shape. Ultimately, I had trouble rendering and texturing all of the modular code and pivoted to produce a deliverable to meet assignment requirements.

If I had more time to go through my code, I would remove all of the vertices from the render loop, and call a draw method unique to the shape classes. Within the shape classes, I would take in parameters to set a size for each shape, given height/width or radius. From there, each shape class would have positioning methods that could transpose the shape to a desired location. This would make the shapes easier to create, and easier to manipulate within the scene. It also allows for less work to move them if the scene changes. My render loop would draw, position, texture, and light these shapes without displaying any vertices within the source file.