# 

# Module 7 (7-1: Project)

# Dallas Weber

# Department of Computer Science

# CS-330-T5525 Comp Graphic and Visualization 23EW5

# Professor: Dr. Kurt Diesch

# June 18th, 2023

For my 3D scene, I chose to recreate my desk at work. The reason I chose this is because it is a scene that I am familiar with and it contained all the necessary objects that were required for this project. The scene contains rectangles that would represent a stack of books on my desk in the back right. When placing the texture for the books, I realized that adding white strips with black dots placed uniformly on the left and right side would mimic the look and covers of the books. A pyramid paper weight shape sits on the top of the books. My large cylindrical sports cup sits on the front left of the scene and a plane which represents the top of my desk are where all the objects are placed. For the rectangles, pyramid, and plane, I specified the specific vertices and indices needed to draw the objects. The cylinder was more complex and as such, I utilized a loop through the number of segments specified which provided for the vertices and indices. All the objects in the scene utilized textures. As such, texture coordinates were used to place the textures in the appropriate location.

Navigation and panning of the 3D scene are done using a keyboard and a mouse. Using the “W” key, the user can move forward in the scene. The “S” key can be used for moving backward. The “A” and “D” keys can be pressed to move left or right. “Q” and “E” perform down and up movement. A toggle of the “P” key switches between a perspective and orthographic view. The scroll wheel on the mouse can be used to move towards and away in the context of the 3D scene while moving the mouse to the left or right causes a panning effect in the direction moved.

The custom functions utilized in the project fall into categories. An initialization function, UInitialize(), is responsible for setting up the environment. This initializes the windowing system, setting up OpenGL and loading resources. The rendering function, URender(), is responsible for drawing the current frame. For handling inputs, I used UProcessInput(), UMousePositionCallback(), UMouseScrollCallback(), and UMouseButtonCallback(), and UProcessInpu. Utility functions such as loading a texture, creating a mesh, or creating a shader program are used throughout the program. For example, the UCreateTexture(), UCreateMeshBook1(), UCreateMeshCylinder(), UCreateMeshPlane(), and UCreateShaderProgram() functions. For cleanup, UDestroyMesh(), UDestroyTexture(), and UDestroyShaderProgram() functions are employed to free resources. The code was organized to initialize, load resources, state setup, render loop and lastly cleanup.