My selected scene of a clear plastic pyramid paper weight, a crayon box, and a Chapstick tube sitting on a red piece of paper made the object choices straight forward for programming. The shape of the crayon box was made of a cube and a plane for the flap, the pyramid paper weight is a pyramid, the Chapstick tube is a cylinder, and the paper they were sitting on a plane. The two point lights were cubes so their information could be reused, just with changing the positions and scales.

A user can navigate the scene with a keyboard and/or a mouse. Keyboard presses were included to move the virtual camera; pressing W moves forward (zooms in), S moves backward (zooms out), A moves left, D moves right, Q moves up, and E moves down. Moving a mouse has the same effect as the keypresses. To make this easier, learnopengl.com’s (Creative Commons) camera class was included, camera.h, I just added up and down as it had left, right, forward, and backward.

Functions that made the program more modularized can be seen in the screenshot below:

Graphical user interface, text, application

Description automatically generated

UCreateMesh will create a VAO, VBO, and number of vertices for an object, mesh was made as a struct and could be reused for multiple objects. UDestroyMesh will release the data from the VAO(s) and VBO(s). UCreateTexture initializes everything needed for an image such as making the image vertical, wrapping parameters, filtering parameters, and determines the channels for the image. UCreateTexture was used for each texture as well as UDestroyTexture to release the data. UCreateTexture made it so this did not have to be done each time I binded a texture to an image:

Text

Description automatically generated

These functions are helpful and could be reused for other programs, GL\_REPEAT could always be changed if a different wrap was wanted. The shaders could also be reused as long as the layout locations are paid attention to when doing attribute pointers. Opengl.com also has shader classes that could be used like the camera class, some would be particularly helpful if one wanted to use multiple types (and amounts) of lights in the project.