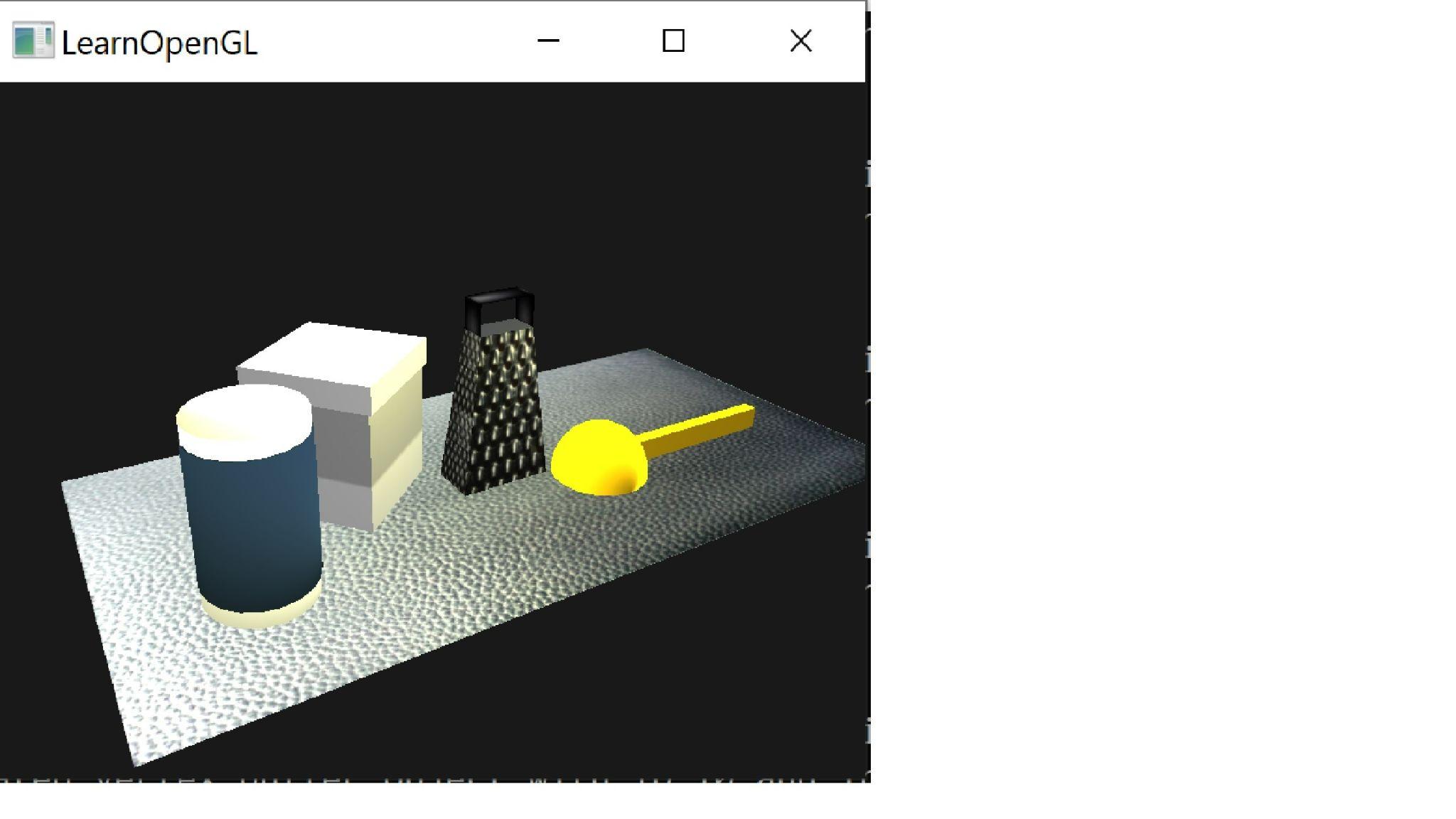
CS 330

Project Reflection

Mitchell Lynds

December 19, 2021



I chose the objects in this scene for several reasons. First, the chosen objects break down easily into primitive shapes such as the cylinder, cube, pyramid, sphere and plane. I picked two objects that would require more complex construction. The cheese grater required the combination of a pyramid and several planes in order to create the handle. The citrus juicer also required the combination of a sphere for the body and a manipulated cube for the handle. In the end I also decided to use multiple objects on the flour container and the salt container as it eased the process of applying textures.

I felt that the variety of shapes would challenge me to learn a wider variety of the course material. In implementing the objects I found I struggled the most with the handle of the juicer. I do think that with more practice and study a more faithful recreation of the handle could be made. Perhaps with a distorted partial torus.

The objects I selected had a variety of materials and surface finishes as well. The Juicer is solid colored and glossy. The salt is multicolored and matte textured. The flour container is made primarily of transparent plastic. Finally, the cheese grater has a very rough surface and is polished. Each object needed to be textured differently to look realistic. I am pleased with the final product of my texturing work on every object except for the flour container. Representing semi-transparent material in openGL is still beyond my knowledge level for now. I feel that the cheese grater best displays a variety of texturing techniques. The narrower sides have the texture tiled to create a smaller hole pattern like the real object. The specular map causes the light to reflect most off of the high spots on the surface.

The 3D scene is navigated using both the keyboard and the mouse. Mouse pointer movement controls the camera direction (pitch and yaw). The mouse scroll wheel adjusts the camera movement speed. The W, A, S, D, Q, and E keys control the panning movement of the camera. W for forward, A for left, S for backward, D for right, Q for up and E for down. Lastly, the P key toggles the camera projection from a perspective view to an orthogonal projection.

A custom function in my program contains the perspective switching code and supplies either the orthogonal or non orthogonal perspective matrix to the render loop. This function could easily be added to future projects making it reusable. There are certainly other parts of my program that could have become modular functions to improve readability and modularity. For instance, many of my objects ended up being based on scaled and translated cubes. A DrawCube function would have cut down on how many lines of code had to reside in the render loop.