The objects I chose to replicate in a 3-D scene are a coffee mug, a desk display item, a trash can, and scrap piece of wood. I chose these objects for their simplicity in design and overall structure, since most of the items were basically a single 3-D shape. I only anticipated that I would need to have a complex designed object, meaning needing multiple shapes to make the object, that object being the coffee mug. I quickly noticed that I would have another complex object, which was the trash can that had a rounded and not perfectly flat bottom. The first object I chose to program was the coffee mug, using a cylinder and torus to complete the basic structure of a mug. The biggest struggle I had in creating the coffee mug was texturing the side of the cylinder itself and trying to get the image to only appear on the side in a certain area. I was able to solve this issue after consulting with my professor who described a way to create a custom texture. I took a generic image close to what appears on the mug and placed it over a black background to match the base color of the mug, ensuring the image looked normal and not “stretched.” The next challenge I had was to give the mug an “open” top in the same way a mug has three sides and not four like a soda can or typical cylinder. After finishing the texturing of the mug, I moved onto trying to solve a similar issue when texturing the office desk item, I was replicating where the image on the front was appearing on all four sides. Again, after discussing with my professor, he encouraged me to explore the Shape Mesh file of the program and I discovered a similar function that drew each side of the mesh instead of the one mesh as a whole. I was able to draw the texture on just the front face of a box mesh, and then use a black texture for the rest of the item so it only showed the image on the front. Creating the trash can wasn’t too difficult as it only involved creating a smooshed half sphere with a cylinder laid on top of it. Getting the two objects to appear seamless just took adjusting the position of the cylinder along the y-axis until it looked to be one item and using some lighting to ensure the object didn’t have a gap between the two objects. I used the same strategy as the coffee mug to give it an open top and some lighting to allow shading to appear so it appeared more open and deeper. Creating the piece of scrap wood was pretty simple since it was a simple box stretched in a rectangle and then applying a wood texture, I found that had a good wood color applied to all of the sides.

Users can navigate the scene using the W, A, S, D keys for simple direction changes of the camera that move it left, right, in, and out. I then added the function of the Q and E keys to move the camera vertically up or down for a different perspective of the items. The mouse itself can be used to move around the scene in a better way than the keys, since it allows the camera to be moved in a 360-type pattern instead of staying in the same perspective of straight ahead, this allows the user to move around the entire scene similar to walking around the objects. Finally, the scroll function of the mouse allows the user to control the speed of the camera where scrolling speeds up the movement speed and scrolling down slows down the movement in the same way.

I created custom functions for each object in the scene to include RenderTableTop, RenderMug, RenderOfficeDeskItem, RenderTashCan, and RenderScrapWood. Combining the different portions of code that were previously generated in RenderScene to create the 3D objects and then trying to use better coding practices using function calls should allow the program to run smoother and more efficiently This would make creating separate scenes with different objects added into it, or to simply take those objects and use them in another program. Maybe if I were to be building something like a video game where objects are used multiple times, like buildings in popular first-person shooter maps they re-use buildings on different parts of the map and just change their position and orientation.