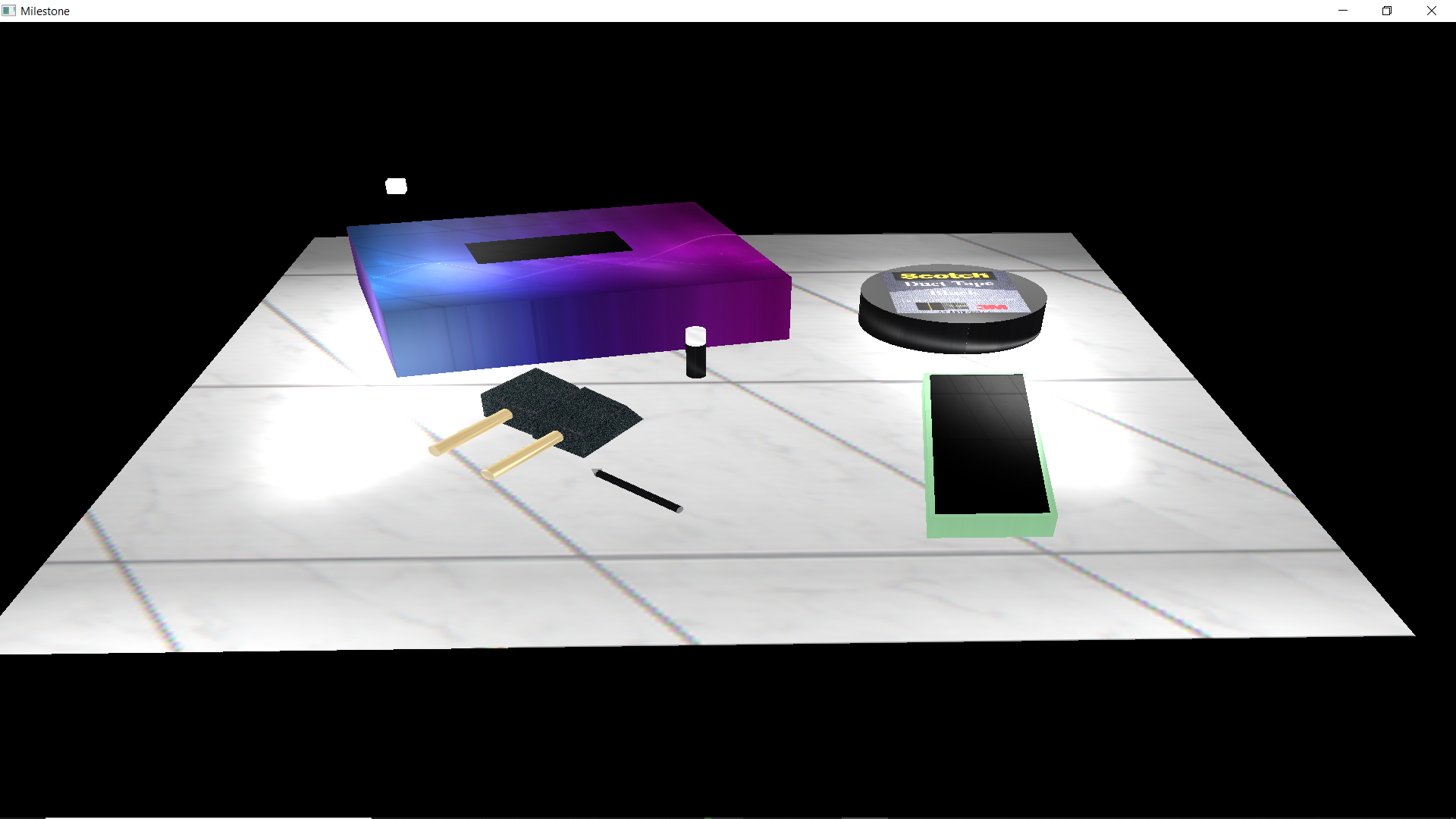
**Justify development choices for your 3D scene**



The reason I chose my selected items was because all of them required multiple shapes in order to create the objects. The cell phone required a cube in order to make the body as well as a plane on top of the cube to display the screen. The duct tape only needed a cylinder, but I placed a plane with a duct tape brand to place on top of the cylinder. The tissue box had a cube for the body and a plane was utilized to create the opening for the tissue box. The foam paint brushes used a trapezoid for the foam heads of the brushes and cylinders for the wooden handles. The pen had a cylinder for the body and a triangle for the tip. Lastly, the chap stick used two cylinders on top of each other for the chap stick’s body and the chap stick’s cap. The only single shape item was the plane with the tile texture because it was the “floor” or kitchen countertop from the 2D picture. When it came to the planes, cubes, and trapezoids, I used vertices and indices to create the shapes and connections to build these objects. For the cylinders, I used the cylder.cpp to establish the formula for these objects. For the cylinders representing the pen and wooden handles, I used a translate and rotation code in order to make the cylinder go onto their side to accurately represent their forms from the 2D photograph.

**Explain how a user can navigate your 3D scene**

For the camera control of the 3D scene, I used many GLFW\_KEY and mouse control to manipulate the movement of the camera system. The key controls A, W, S, D, Q & E were used for the body movement of the camera and the mouse control was used to control vision movement of the camera. Comparing this to a body, the GLFW\_KEYS would be considered a body walking forward, backwards, side to side, up, and down. The mouse control would be considered as an owl being stationary while they turn their head to look in different directions. The zoom control for the mouse control allows a user to zoom back and forth to get close or away from an object. The last GLFW\_KEY I utilized was P and it enables the user to view the scenario in a 2D manner. If the user holds down the P key, they can continue to see the scenario in a 2D world. However, once they let go of the P key, the scenario goes back into a 3D scenario.

**Explain the custom functions in your program that you are using to make your code more modular and organized**

There are multiple custom functions in my code program. The first custom functions were in the vertices and indices. For the trapezoids representing the foam heads, I had two vertices points in the same location to affect the texture of the heads. Looking at my code, the two points are the same except that they each have different texture ranges. These different texture ranges allowed me to do the brush heads rough foam texture, the top of the brush heads, and on the sides of the brush heads. The custom function was the switchProjection, which I set as a bool in order to give the user the option to view the scenario in a 2D or 3D world. The last custom function was adding a glm::translation and glm::rotation to multiple cylinders within my scenario. I was able to move the cylinder to the exact location on the x, y, and z ranges using the glm::translation command. After that, I was able to rotate the cylinders in a 90-degree angle using the glm:rotation command to place the cylinder on their sides and accurately represent the 2D photograph. These codes are all reusable to either create the same scenario or to make changes for a different scenario. If changes are made to these codes for different scenario creations, the numbers for the ranges, vertices, and indices would have to be changed to manipulate the design and movement of the objects.