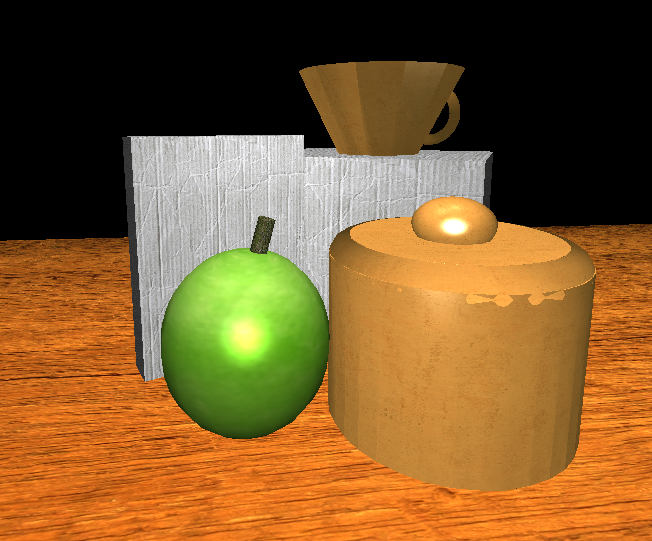
David King

6/12/2024

CS 330

Final Project Design Decisions

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



The above scene was chosen due to the variety of items available. Various orientations, materials, degrees of shininess, and colors are present within the picture. These variations provided a good opportunity to showcase how different elements of visual design can be utilized to create distinct objects. When building the scene, material properties such as surface shine were considered. Items such as the apple and ceramic pieces were shinier to reflect their real-life counterparts, whereas the cardboard boxes do not reflect anywhere near the same level of light. This contrast is also visible between the apple’s stem and the main body of the fruit. Similar considerations were made with colors, where a more vibrant, green color was used for the apple and an earthy beige was employed for the ceramic pieces.

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

Scene navigation is performed using the WASD keys for forward, leftward, backward, and rightward movements respectively. The Q and E keys are used to move up and down. The camera angle is adjusted via the mouse. The scroll wheel can be used to adjust the speed of both camera positioning and camera turning. Finally, the O key toggles the orthographic view, whereas the P key toggles a porthole view.

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

Most of the custom code added to the project file revolves around defining and implementing tools such as textures, shader materials, and lighting effects. As an example, DefineObjectMaterials held the definitions of object material properties in respect to how those objects interacted with the scene’s lighting. Factors that effect these interactions include things like material shininess, ambient color, ambient strength, diffuse color, and specular color. These definitions interfaced with adjacent light definitions from SetupSceneLights such as ambient color, diffuse color, and specular color. Every light calculation takes both sets of values into account to produce a different result from item to item. The consequence of this method is that some objects can appear shinier than others, or otherwise interact with the lighting differently.

The PrepareScene method was the largest aspect of the custom code by far, holding values and definitions that dictated what textures, shaders, orientations, and shapes effected every item.