Jimar Mims

CS 330 Comp Graphic and Visualization

Professor Battersby

February 23, 2025

Final Project Reflection

For my 3D scene, I chose four objects that are commonly found on a desk: an orange, a cologne bottle made of two cylinders (a larger one for the body and a smaller one for the cap), a light blue tongue scraper package (a rectangle), and a hair gel container (a cylinder). These objects were selected because they represent a mix of organic and geometric shapes, which provide a great opportunity to test lighting and texturing techniques. The cologne bottle, with its distinct shape, allowed me to explore the challenge of rendering a multi-part object, while the orange gave me a chance to simulate more complex lighting effects on an organic, textured surface. The tongue scraper package and hair gel container are simple yet effective for understanding how light interacts with different materials. The arrangement of the objects, from left to right orange, cologne bottle, tongue scraper package, and hair gel container was chosen for visual balance, ensuring that each object could be individually highlighted by the lighting.

To enable navigation within the scene, I set up keyboard and mouse controls for the virtual camera. Users can rotate the camera using the mouse to look around and use the keyboard’s WASD keys to move the camera forward, backward, and along the X and Y axes. Users can also navigate up and down using the Q and E keys respectively. This setup provides an intuitive way for users to explore the 3D environment. I also made sure that the controls were responsive by adjusting the sensitivity of the mouse for smooth rotation and using a constant speed for the keyboard-based camera movement to avoid abrupt transitions.

To keep the code modular and organized, I developed custom functions like setupLighting() for configuring the light sources and updateCameraPosition() for controlling the camera’s movement based on user input. The setupLighting() function allows me to easily adjust the lighting properties, such as intensity and position, making it reusable in other scenes. Similarly, the updateCameraPosition() function handles all camera translations and rotations, and it can be reused whenever a scene requires dynamic camera control. By breaking down these functionalities into distinct functions, I created a more flexible and maintainable codebase that can be easily expanded or modified for future projects. Whether that’s changing keys to fit different projects and camera functionality or adding new keys for new functionalities. Also, although I never fully got the lighting to work in my scene the base code is present.