Binaya Rimal

10/19/2024

CS 330 – Final Project

The 3D scene I created is a digital representation of my computer desk. It includes several items: a vase, a roll of paper towels, a monitor, and a cup. I selected these objects to challenge myself with varying levels of complexity, as each item required different modeling techniques to capture its shape accurately.

* **Roll of Paper Towels:** Modeled using a cylinder for the body and a sphere on top, making it the simplest object in the scene.
* **Cup:** Constructed with a cylinder for the body, a small sphere at the lip, and a half-torus for the handle, adding more complexity.
* **Vase:** A blend of a sphere, cone, and cylinder, providing practice with smooth shape transitions.
* **Monitor:** Built using box meshes of varying sizes to represent the screen, stand, and frame, requiring precise proportions.

All objects rest on a box mesh representing the desk surface to give a 3d perspective, and a plane mesh representing the wall in the background to show accuracy to the original image.

To navigate the scene, users can adjust the camera’s perspective by moving the mouse. The following keys allow users to move towards the specified directions: ‘A’ to move left, ‘D’ to move right, ‘Q’ to move upward, and ‘E’ to move downward. Users can move forward by pressing ‘W’ and backward by pressing ‘S’, making it easy to explore the space from different angles and depths. Users can also scroll the mouse wheel to adjust the movement speed, allowing users to slow down for careful exploration or speed up for fast navigation. This control scheme ensures smooth and intuitive interaction within the scene.

To create a modular program, I separated each item into its own function. These functions include createPaperTowel(), createFlowerPot(), createDesktop(), createCoffeeCup(), createTable(), and createWall(). Each function combines multiple shapes to build the desired object. All these items are called from the renderScene() function to assemble the final scene.

Some objects require textures, which are handled in the loadSceneTexture() function. This function loads images from the texture folder by specifying their paths and integrating them into the project. Additionally, each object has unique material properties, such as gradients and shininess, defined in the defineObjectMaterial() function. For example, a wooden object has a duller finish, while a metal object appears more reflective.

To enhance the scene’s depth, we also incorporate lighting. There are four light sources, each configured in the setupSceneLights() function, with a warmer tone to match the lighting of the original scene.

Overall, texture application, material properties, and lighting add realism, and interactive camera controls enhance user engagement. If I had more time and expertise, I would have liked to add flowers inside the vase. However, due to limited time and knowledge, that wasn’t possible. The most challenging part of the project was achieving a seamless transition between the shapes in the vase. Limitations aside, the 3d rendering create a mostly accurate representation of the image that it was based on.