Final Project

Valeria Duharte

CS330: Comp Graphic and Visualization

Professor Omar Aaziz

August 2025

For my 3D scene, I based the environment on a real-life reference: a dark coffee table in my living space. I chose to replicate personal objects that reflect my daily routine, this made the scene both meaningful and realistic. Each object served a unique modeling challenge:

* **Laptop**: Chosen for its simple box structure, the laptop was modeled using two aligned and rotated rectangular boxes. It represents a device I use daily and anchors the scene visually.
* **Spiral Notebook**: The notebook was created from a box shape with small **torus** loops along the side to simulate the metal spirals.
* Mug: A **cylinder** represents the mug sitting on top, and I added a **half torus** to model the handle, which was slightly challenging to align correctly without twisting.
* **Star Stress Ball**: This was the most complex object to model. It was constructed using a **sphere** as the core, with five **cones** carefully placed around it to form the star’s points. Aligning and scaling the cones to look proportional required trial and error and spatial calculations.

By focusing on accurate proportions and logical object placement, the final scene feels balanced and true to the reference image. To manage complexity and maintain readability, I developed custom functions to modularize my code:

* DefineObjectMaterials() – This function allows me to define and reuse multiple material types (e.g., gold, rubber, cement). Each material has defined diffuse, ambient, specular, and shininess values, enabling a polished visual presentation with Phong shading.
* SetupSceneLights() – This function initializes all light sources, including directional, point, and spotlight types. By centralizing light configuration, I can easily test and adjust brightness, color, and positioning without cluttering the main loop.
* PlaceObject(name, position, rotation, material) – A helper used to position and render each 3D object with its corresponding transformations and materials. This makes the code more scalable and easier to maintain.

These functions are **reusable** and **modular**, allowing the addition or removal of objects or lights without rewriting core logic. The 3D scene is fully navigable using keyboard and mouse inputs:

* Keyboard Controls: W / S: Move camera forward and backward. A / D: Pan camera left and right. Q / E: Move camera vertically up or down
* Mouse Controls: Mouse movement: Rotates the camera view (look around). Mouse scroll: Controls the movement speed, letting users navigate smoothly

This combination allows users to explore the scene freely and inspect details from different angles. It also simulates real-world movement in a 3D space, providing an intuitive experience.

To meet visual and functional goals: A **warm directional light** simulates natural sunlight from above-right, softening shadows. A **blue-colored point light** above the cylinder/torus area creates vibrant highlights. Both lights have full **ambient, diffuse,** and **specular** components.

Each light includes ambient, diffuse, and specular components. Their placement ensures all objects are well-lit, even when the camera rotates around them. The lighting not only highlights object shapes and textures but also enhances the realism of the shadows and reflections.