**3D Scene, Final Project**

**CS 330 – Jorge Cisneros Rodriguez**

**SNHU**

**Reflection**



I chose a compact desktop vignette because it maps cleanly to the “few basic shapes” requirement while still letting me demonstrate texturing, lighting, and modular mesh generation. The scene includes:

* **Table**: Planes/boxes with cylinders for legs
* **Monitor**: Boxes for body, screen, chin badge, and stand
* **Keyboard**: Textured plane over a low box
* **Mouse + buttons**: Scaled sphere pieces on a textured mouse pad (thin box)
* **Pencil holder**: Hollow tube with bottom disc, with three full pencils (cylinders + cones + shader-tinted tips)
* **Book stack**: Three rotated books (page block + two covers + spine strip)
* **Mug**: Hollow tube with a curved handle (torus section) and liquid (flat interior disc)

All models are low-polygon approximations built from primitives. Where curved silhouettes were important (holder/mug walls, handle), I used modest segment counts (e.g., tube segments = 48; torus section U/V = 96/32; cones ~32; sphere from the starter meshes) so each object stays well under 1,000 triangles but still shades smoothly under Phong lighting. Reuse is a theme: the same tube mesh draws both the pencil holder and the mug; cones are reused for pencil tips; boxes/planes are reused for the table and monitor.

Textures are applied where they add the most realism per triangle: wood for the tabletop, a keyboard photo mapped onto the top plane of the keyboard housing, a mousepad image on a thin box, and a wood texture for the pencil tips. UVs from the shared meshes keep projection simple and free of stretching.

For lighting, I combined:

* A directional light for broad fill and soft shading
* A point light with warm tint to add localized highlights and depth

Shaders implement ambient, diffuse, and specular components with controllable shininess/specular strength per object, which lets metallic/plastic pieces read differently than paper/wood.

Placement matches a typical desk layout and the client image: monitor centered rear, keyboard in front, mouse and pad at the near-right corner, pencil holder to the monitor’s right, book stack at the front-left, and the mug at the rear-left. Affine transforms (translate/rotate/scale) are kept local so each object can be nudged without side effects.

**Scene navigation and controls**

The camera supports 6-DOF translation and mouse-look:

* **W/A/S/D**: Forward/left/back/right
* **Q/E**: Down/up
* **Mouse movement**: Yaw and pitch (with clamped pitch to prevent flipping)
* **Mouse scroll**: Adjusts movement speed at runtime for fine or fast travel

Movement is frame-rate independent via deltaTime. The orbit/capture radius is large enough to keep the entire desk lit and visible from multiple angles. There is also a single-key projection toggle that switches the active projection matrix between perspective and orthographic while preserving the current camera orientation, which is helpful for verifying proportions and alignments.

**Custom functions and modularity**

To keep the code organized and reusable, several focused helpers are used:

* **BuildTubeMesh / DrawTubeMesh**: Generates a hollow cylinder with an open top and a bottom disc. One VAO serves both the pencil holder and the mug; scale and placement produce the variations.
* **BuildTorusSectionMesh / DrawTorusSectionMesh**: Produces a curved handle as a torus arc with adjustable major/minor radii and start/end angles. This creates a smooth handle that can be positioned and scaled for different cups if needed.
* **Per-object material setters (via ShaderManager)**: Centralizes toggles such as useTexture, colorTip, and uniform updates for colors, tinting of pencil tips, shininess, and specular strength.
* **DrawBook**: A small builder that composes one book from covers, page block, and spine strip, so the three-book stack can be varied by color, dimensions, yaw, and center position.

This modular approach avoids duplication, keeps meshes reusable, and makes it straightforward to add or replace assets later without rewriting core logic.