

**Final Project Reflection: 3D Scene Design, Controls, and Modularity**

Dave Mobley

Southern New Hampshire University

CS-330

Eugenio Rodriguez

August 24, 2025

## Final Project Reflection: 3D Scene Design, Controls, and Modularity

### Justification and Object Choices

The scene focuses on a glass a saucer, and a straw because these items map cleanly to low-polygon primitives while still allowing meaningful demonstrations of texturing, lighting, and shadowing. The mug is constructed from two tapered cylinders (an outer shell and an inner liner) so that the rim reads correctly and the walls have plausible thickness. The saucer combines a thin cylinder with a shallow rim ring, which together present as a plate. The liquid surface is modeled as a single top cap (for performance efficiency) and enhanced in the shaders with subtle ripple and meniscus cues. A slender cylindrical straw provides a contrasting vertical element and a useful test for shading interactions through the liquid.

### Functionality and Architecture

Responsibilities are separated to keep the code modular and testable. The **ViewManager** handles camera state, input processing, per-frame timing, and projection mode switching. The **SceneManager** owns transforms, materials, textures, lighting configuration, and all draw calls, including a dedicated routine for spotlight shadow mapping. The shader programs implement Phong lighting (ambient, diffuse, specular), a directional light for overall fill, a dim point light to avoid fully dark regions, and a camera-tied spotlight that produces focused illumination and shadows.

### Navigation and Camera Control

Keyboard controls support full six-degree translation of the camera: **W/A/S/D** move forward/left/back/right, **Q** or **Space** move up, and **E** or **Left Ctrl** move down. The **P** key switches to a perspective projection, and the **O** key switches to an orthographic projection while keeping the camera orientation fixed. Mouse motion adjusts yaw and pitch for nuanced viewing, and the mouse scroll wheel adjusts camera movement speed. Acceleration and deceleration are applied for smooth navigation. The camera's position and direction

are also used to drive a flashlight-style spotlight each frame.

### Lighting and Materials

Lighting is staged to balance clarity and realism: a directional light provides broad fill, a dim point light prevents hard black areas, and a camera-tied spotlight adds emphasis and shadows. Materials use Phong parameters with tuned shininess and specular response. The saucer is configured to read as an opaque, reflective ceramic/glass surface by using strong specular highlights with high shininess.

### Modularity and Reusable Functions

- **SceneManager::SetTransformations(scale, Xdeg, Ydeg, Zdeg, pos)**  
centralizes model matrix construction (scale, rotate, translate) and is reused for every object, simplifying layout changes.
- **SceneManager::SetShaderColor(r, g, b, a)** and **SetShaderTexture(tag)**  
switch between color and texture flows in a uniform way, avoiding duplicate boilerplate.
- **SceneManager::SetTextureUVScale(u, v)** exposes texture tiling without modifying shader code.
- **SceneManager::SetShaderMaterial(tag)** decouples material selection and can be extended into a reusable catalog.
- **SceneManager::RenderShadowMap(lightPos, lightDir)** encapsulates the depth pass and mirrors scene transforms so shadow updates remain isolated.
- **ViewManager::ProcessSmoothMovement(), ProcessProjectionKeys(),** and **ProcessRippleControls()** keep input concerns separated and easy to adjust.

### Performance and Triangle Budget

All meshes are low-polygon primitives so that each object is kept well under the 1,000-triangle target. Texture sampling uses mipmaps. The spotlight's shadow map is

rendered at  $2048 \times 2048$  to balance crisp contact shadows against runtime cost.

### **Primitives Summary**

Table 1 summarizes the objects, their underlying primitives, and counts.

### **Notes**

Cylinder disks are rendered with the cylinder mesh by enabling only the top or bottom cap. The simple set of primitives keeps the scene performant while achieving the intended visual design.

**Table 1***Primitive Shapes Used in the Scene*

<b>Object</b>	<b>Primitive</b>	<b>Count</b>
Table surface	Plane	1
Saucer base	Cylinder	1
Saucer rim ring	Cylinder	1
Mug outer shell	Tapered cylinder	1
Mug inner shell	Tapered cylinder	1
Liquid top cap	Cylinder (top cap only)	1
Mug bottom cap	Cylinder (bottom cap only)	1
Straw	Cylinder	1
<b>Totals</b>	Primitive draws	<b>8</b>