**Final Project – Final Scene Reflection**

[**CS-330**](https://learn.snhu.edu/d2l/home/2019696)[**Comp Graphic and Visualization**](https://learn.snhu.edu/d2l/home/2019696)

**Daryl Murtha**

**October 16, 2025**

**Development Justification and Technical Analysis**

**Scene Object Selection Rationale**

I chose to create a modern office workspace scene to provide a realistic, relatable environment that naturally incorporates diverse geometric primitives while telling a cohesive visual story. Each object contributes to the scene’s functionality, lighting complexity, and spatial realism:

* Desk Surface (Box): Establishes scale and context with a warm wood texture and shadow-receiving geometry.
* Laptop Assembly (Multiple Boxes): Central focal point with base, angled screen, display panel, and 60 individually lit keyboard keys.
* Coffee Mug (Cylinder + Torus): Curved geometry with ceramic material and dynamic rotation for shadow interplay.
* Desk Lamp (Cylinder + Cone): Multi-primitive object with cone geometry and dual textures (gold shade, steel base).
* Floor and Wall (Boxes): Environmental backdrop with tiled UV scaling and shadow projection surfaces.

**Navigation System Design**

To support full 3D exploration, I implemented a multi-input control scheme:

* Keyboard Controls: WASD for movement, Q/E for vertical traversal, P/O for projection toggle, ESC for exit.
* Mouse Controls: Yaw/pitch orientation, scroll-based speed adjustment (0.1× to 10×), cursor capture for immersion.
* Frame-Rate Independence: Delta time calculation ensures smooth movement across hardware variations.

float currentFrame = static\_cast<float>(glfwGetTime());

gDeltaTime = currentFrame - gLastFrame;

gLastFrame = currentFrame;

**Projection System Innovation**

Users can toggle between orthographic and perspective views at runtime:

if (bOrthographicProjection) {

projection = glm::ortho(-10.0f, 10.0f, -10.0f, 10.0f, 0.1f, 100.0f);

} else {

projection = glm::perspective(glm::radians(g\_pCamera->Zoom), aspectRatio, 0.1f, 100.0f);

}

This supports both technical analysis and immersive realism, enhancing spatial understanding and lighting evaluation.

**Custom Function Architecture**

**Modular Draw Functions**

Each object is rendered via a dedicated function (e.g., DrawLaptop(), DrawCoffeeMug(), DrawDeskLamp()), using centralized constants for transformation, texture, and material configuration. This structure supports reusability and clean separation of logic.

**Material Management System**

The SetShaderMaterial() function configures ambient, diffuse, specular, and shininess values based on texture tags:

* Wood: Warm tones, moderate shininess (32.0)
* Metal: Neutral colors, high shininess (256.0)
* Ceramic: Cool tones, very high shininess (128.0)
* Stone: Matte finish, low shininess (64.0)

Modularity: Materials are defined once and applied consistently, supporting easy expansion and tuning.

void SceneManager::SetShaderMaterial(std::string materialTag) {

if (materialTag == SceneConstants::LAMP\_SHADE\_TEXTURE) {

m\_pShaderManager->setVec3Value("material.ambientColor", glm::vec3(1.0f, 0.8f, 0.4f));

m\_pShaderManager->setFloatValue("material.shininess", SceneConstants::METAL\_SHININESS);

// Additional PBR properties...

}

// Other material configurations...

}

**Transformation Pipeline**

The SetTransformations() function applies scale, rotation, and position using TRS order:

glm::mat4 modelMatrix = translation \* rotationX \* rotationY \* rotationZ \* scale;

m\_pShaderManager->setMat4Value(g\_ModelName, modelMatrix);

Reusability: Used by every object in the scene, ensuring consistent placement and orientation.

**Advanced Lighting and Shadow System**

**Dual Light Configuration**

* Primary Spotlight: Dramatic lighting for coffee mug shadows.
* Secondary Accent Light: Blue-tinted point light for visual contrast.

Innovation: Satisfies rubric requirement for a second light source with distinct color. The accent light introduces a cool-toned contrast to the warm spotlight, enhancing depth and visual interest.

light.position = SceneConstants::DRAMATIC\_LIGHT\_POSITION;

light.spotDirection = SceneConstants::DRAMATIC\_LIGHT\_DIRECTION;

light.isSpot = true;

light.position = SceneConstants::ACCENT\_LIGHT\_POSITION;

light.diffuseColor = SceneConstants::ACCENT\_LIGHT\_DIFFUSE;

light.isSpot = false;

**Shadow Mapping Pipeline**

The RenderDepthPass() function renders all shadow-casting objects from the light’s perspective using a depth shader:

void SceneManager::RenderDepthPass(glm::mat4 lightSpaceMatrix) {

glBindFramebuffer(GL\_FRAMEBUFFER, m\_shadowMapFBO);

glClear(GL\_DEPTH\_BUFFER\_BIT);

m\_pDepthShaderManager->use();

// Render all objects including new desk lamp

}

Modularity: Separate depth pass enables high-quality shadow mapping for all objects with consistent transforms via SetModelMatrix().

**Technical Innovation Highlights**

**Constants-Based Architecture**

All transforms, materials, and configurations are centralized in SceneConstants, eliminating magic numbers and supporting easy scene modification.

namespace SceneConstants {

// All object transforms, materials, and configurations

// Centralized for easy modification and maintenance

}

Benefit: No magic numbers; easy scene modification; professional code organization.

**Dual Shader System**

* Main Shader: Full PBR rendering with lighting and shadows.
* Depth Shader: Optimized for shadow map generation.
* Consistent Transforms: SetModelMatrix() applies to both shaders simultaneously.

**Advanced Material System**

Each material type is tuned for realistic surface behavior and lighting response:

* Wood: Warm tones, moderate shininess (32.0)
* Metal: Neutral colors, high shininess (256.0)
* Ceramic: Cool tones, very high shininess (128.0)
* Stone: Matte finish, low shininess (64.0)

**Final Reflection**

This architecture creates a professional-grade 3D visualization system that exceeds academic expectations while maintaining clean, modular, and reusable code. Every component from camera controls to lighting and shadow logic is designed for clarity and performance.