**Design Decisions for 3D Scene Rendering Project**

**Overview**  
This project involved designing and rendering a 3D scene that realistically replicates a tabletop workspace using OpenGL and C++. The purpose was to apply transformation matrices, lighting, shading, and texture mapping techniques to create a detailed, textured environment. Key components of the final scene include a round wooden coffee table, a water bottle with cap and label, a coaster, a laptop (keyboard base and angled screen), and a rug. The model reflects a balance between technical OpenGL implementation and visual realism.

**Object and Mesh Choices**  
The scene utilized a variety of primitive meshes: planes, cylinders, boxes, spheres, and cones. For example, the coffee table and bottle body used DrawCylinderMesh() due to their round symmetry, while the laptop and screen were constructed using DrawBoxMesh() to create their rectangular, 3D appearance. A cone mesh was applied for the transition between the bottle’s neck and cap. The coaster was built as a thin cylinder to simulate a round ceramic coaster with depth. These decisions were made to reflect both accurate object proportions and the limitations of available primitive shapes.

**Texture Mapping**  
Texture realism was a critical design goal. Custom textures were applied to key surfaces: a wood grain texture for the table, a custom-designed rug texture, a marble-like image for the coaster, and separate keyboard and screen textures for the laptop. These textures were created and resized to safe OpenGL-compatible dimensions (1024x1024 where appropriate) to prevent loading errors. Each texture was loaded using a function that supports flipping, mipmapping, and variable channel detection. The decision to create custom textures stemmed from a desire to match real-life references and increase scene immersion.

**Lighting and Material Considerations**  
The lighting setup included a single warm overhead point light placed above the center of the table. This provided a soft, ambient glow across the wood surface and created subtle highlights on reflective surfaces such as the water bottle and laptop. The material.diffuseColor, specularColor, and shininess parameters were customized for a balanced level of realism without oversaturation. Default materials were used uniformly across most objects to simplify material tracking while ensuring consistency.

**Camera and Scene Transformations**  
Each object was scaled, rotated, and translated manually using transformation matrices to align it accurately in the scene. For example, the laptop screen was rotated around the X-axis to lean back at a realistic angle. The coaster and water bottle were elevated above the table’s surface by adjusting the Y-axis translation. Object positioning was iterated multiple times with trial-and-error to match reference imagery, accounting for perspective distortion in the final render.

**Debugging and Problem Solving**  
The project encountered a critical issue when a JPEG file (laptopwindow.jpg) caused a segmentation fault due to size or encoding problems. This was resolved by inspecting the image’s properties and resizing it to 1024x1024 to comply with OpenGL’s texture handling limits. Additional error-handling code was inserted to catch missing textures and array overflows, which made the system more robust.

**Conclusion**  
This project demonstrated how a simple scene can showcase advanced OpenGL techniques such as texture mapping, lighting, object composition, and matrix transformations. Each design decision—from object mesh selection to texture resolution and positioning—was guided by both visual goals and technical constraints. Future improvements could include normal mapping, real-time shadows, or user-controlled camera navigation.