# Manejo de Librerías:

* GLFW para ventana/input.
* Se utilizan GLFW para crear la ventana de la aplicación, gestionar el contexto de OpenGL y procesar las entradas del teclado (KeyCallback) y el mouse (MouseCallback
* Toda la manipulación de posiciones, rotaciones y proyecciones se realiza con GLM, que les proporciona las herramientas para trabajar con vectores y matricesClases Model para cargar .obj y sus materiales/texturas.
* Shaders GLSL separados (modelLoading, lighting, lamp, skybox).
* Skybox (cubemap), VBO/VAO para cubos y geometría simple.
* Un sistema de cámara libre (clase Camera) con deltaTime y controles WASD + mouse.
* Un sistema simple de keyframes e interpolación para animación.

# Manejo de Modelos y Texturas:

**Carga de Modelos:**

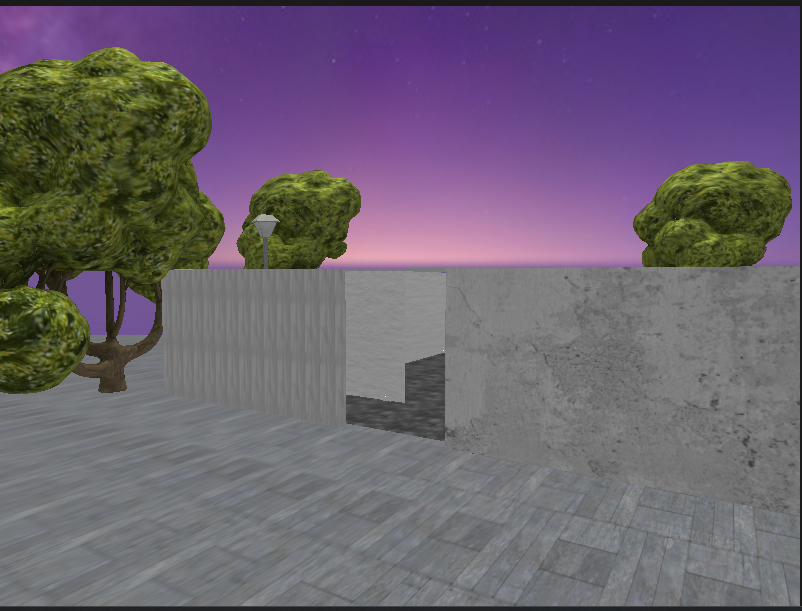
Se utiliza la clase Model para crear instancias, tanto de los objetos utilizados en el escenario como el propio uso de espacios como la galería.

Model encapsula Assimp: lee mallas, materiales, texturas y construye VAOs/VBOs por malla.

Se utilizan archivos en formato .obj para geometría y archivos jpg/png para la aplicación de texturas difusa y specular.

Para la parte de carga de texturas se utiliza las cargas de cubemaps con TextureLoading::LoadCubemap(faces).

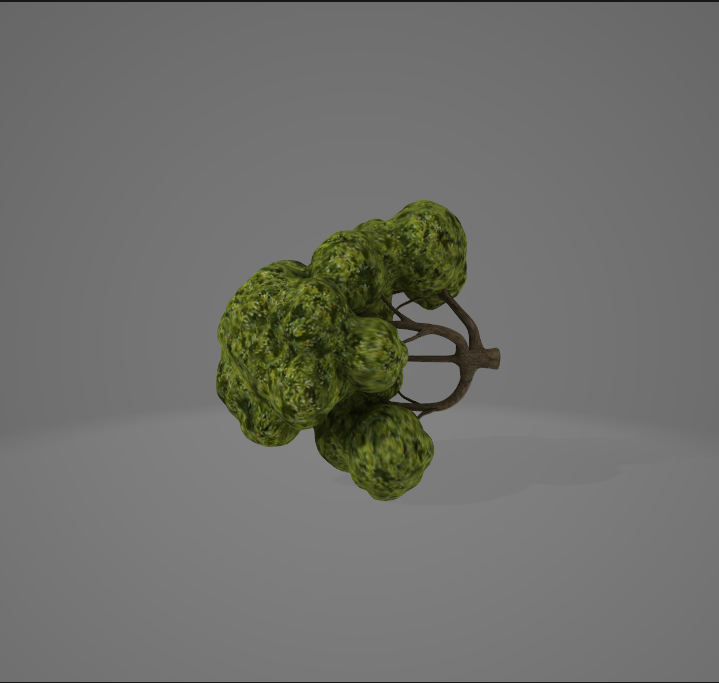
Para la composición de la escena se construye la escena en el bucle principal posicionando cada modelo de forma independiente. Se utilizan transformaciones de traslación (glm::translate), rotación (glm::rotate) y escalado (glm::scale) para colocar cada objeto en su lugar, como se ve en el posicionamiento de los árboles y las lámparas.



# Modelos utilizados:

Los modelos utilizados son primero modelados en blender, después exportados y agregados a OpenGL con el uso de los archivos .obj.

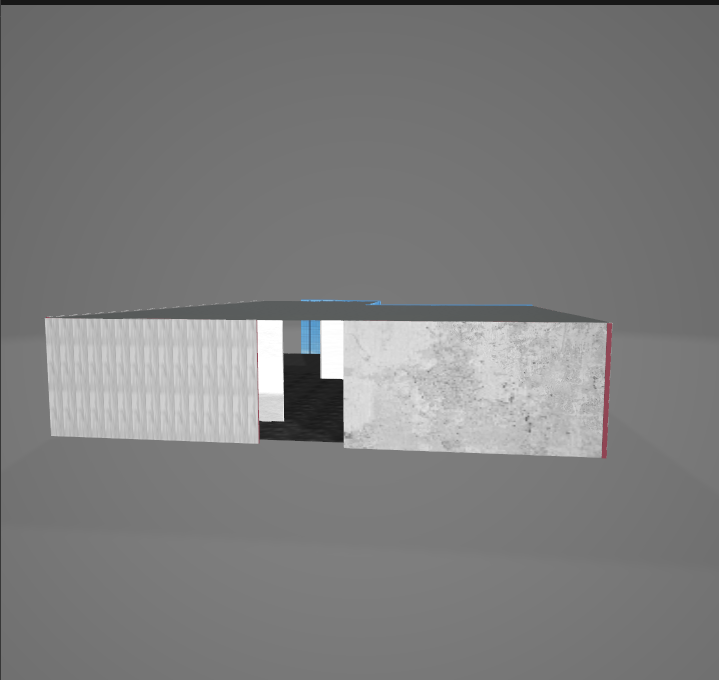
Los modelos utilizados son arboles:



Perros:



Farolas:

El piso y la misma galería:

# Técnicas de Iluminación: