UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

EJERCICIOS DE CLASE Nº 03

NOMBRE COMPLETO: Carlos Alberto Arroyo Ramìrez

Nº de Cuenta: 320185865

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 09/02/2025

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa

El ejercicio consistía en realizar una casa en 3D a partir de geometrías ya hechas: pirámide cuadrangular, cubo, cilindro y cono. Lo que hice fue organizar mi programa en el main e ir insertando bloques de código muy similares entre las piezas, modificando la escala, la posición, el color y el tipo de geometría que usaría.

Para el tipo de geometría se colocaba el índice según lo que se necesitaba, pero también dependía de cuál se iba a insertar. Por ejemplo, para el árbol se requería un cilindro y un cono, los cuales solo están en RenderMeshGeometry; en cambio, los cubos únicamente necesitaban llamar a RenderMesh, y en el caso particular de la esfera se codificaban con sp.render();.

```
//************************Izquierda*****************************************

//Ventana delantera
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-1.75f, 3.75f, -0.75f));
model = glm::scale(model, glm::vec3(1.0f, 1.4f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
```

Ejemplo con RenderMesh.

```
//************************Arbol der*****************************************

//Cono verde
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(4.0f, 3.0f, -1.5f));
model = glm::scale(model, glm::vec3(0.5f, 2.0f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[3]->RenderMeshGeometry();
```

Ejemplo con RenderMeshGeometry.

```
//************************Esfera trasera*****************************************
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 2.8f, -3.0f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(0.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render();
```

Esfera.

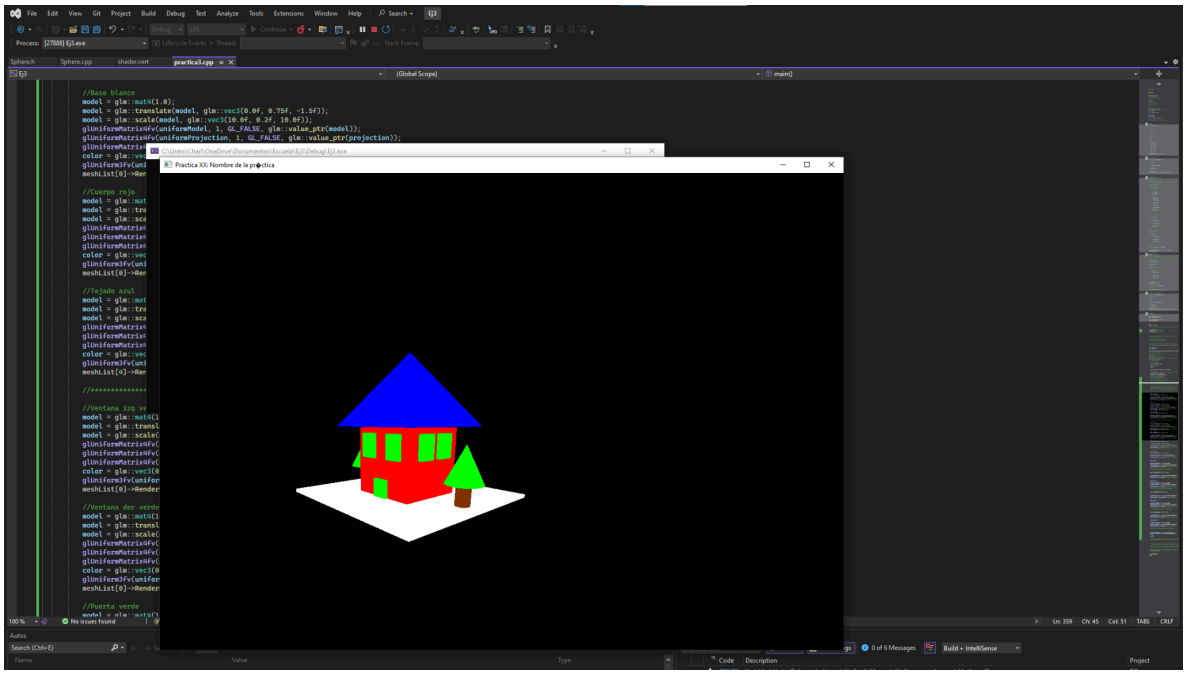
Por último, para poder observar a través de la cámara, solo modifiqué el shader.vert: descomenté gl_Position=projection*view*model*vec4(pos,1.0f); y comenté la línea de código que estaba por debajo.

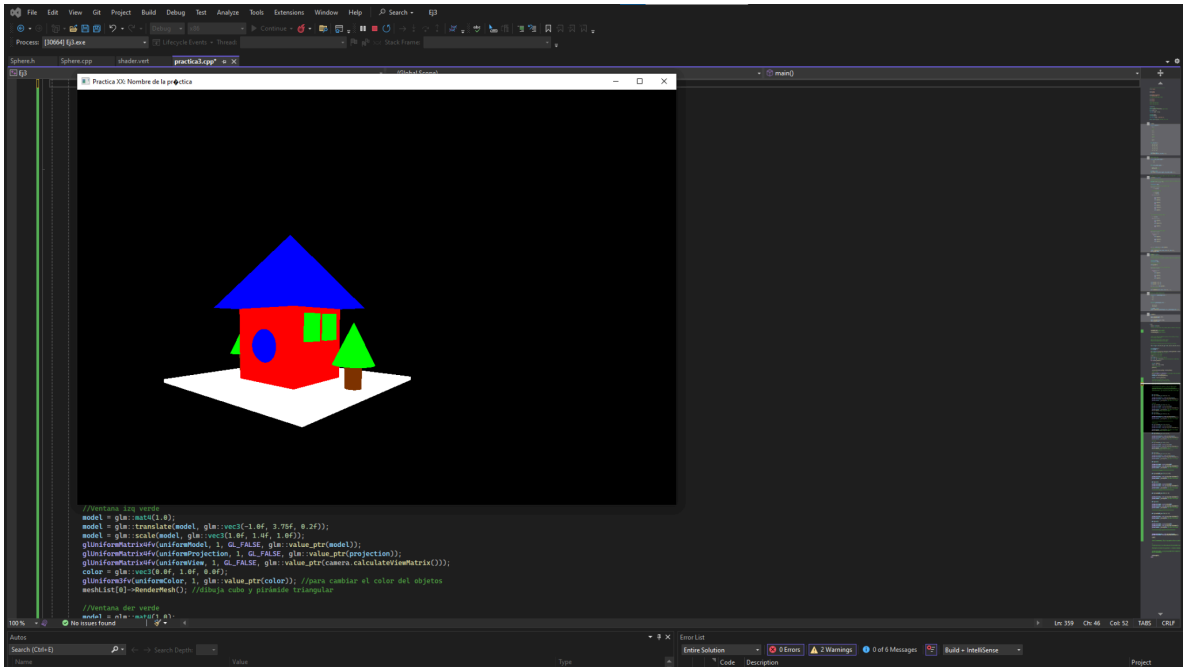
Sphere.cpp | shader.vert | practica3.cpp

```
#version 330
layout (location =0) in vec3 pos;
out vec4 vColor;
uniform mat4 model;
uniform mat4 projection;
uniform vec3 color;
uniform mat4 view;
void main()
{
    gl_Position=projection*view*model*vec4(pos,1.0f);
    //gl_Position=projection*model*vec4(pos,1.0f);
    vColor=vec4(color,1.0f);

}
```

Al final logre los siguientes resultados:





2.- Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código.

Tengo problemas en mi computadora al insertar archivos, a veces no responde y no deja agregar elementos existentes, pero es un problema interno del sistema de archivos, por lo tanto se me complica iniciar un proyecto desde 0, después para esta práctica en particular no tuve ningún problema.

3.- Conclusión:

a. Los ejercicios de la clase: Complejidad, explicación
b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

En esta práctica aprendí a organizar código para insertar distintas figuras geométricas en mi programa, ajustando su posición, escala y color para modelar una casa en 3D. También comprendí la importancia de modificar el shader y controlar la cámara para lograr una mejor visualización de la escena. Aunque se me dificulta crear los proyectos debido a un fallo interno en mi computadora, resultó muy útil contar con el movimiento libre de la cámara, ya que facilita observar y modelar desde diferentes perspectivas objetos básicos como una casa.