



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 N° 3

NOMBRE COMPLETO: Resendiz Casas Axel Ariel

N° de Cuenta: 316208516

GRUPO DE LABORATORIO: 2

GRUPO DE TEORÍA: 4

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 27/08/2024

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas.

La actividad realizada en clase fue replicar la casa del ejercicio en clase 2, solo que ahora debemos hacerlo en 3D, a continuación se muestran los pedazos de código que modifique.

Primero cambié los parámetros del cono.

```
int main()
{
    mainWindow = Window(800, 800);
    mainWindow.Initialise();
    //Cilindro y cono reciben resolución (slices, rebanadas) y Radio de circunferencia de la base y tapa

    CrearCubo();//índice 0 en MeshList
    CrearPiramideTriangular();//índice 1 en MeshList
    CrearPiramideCuadrangular();//índice 2 en MeshList  resolución,radio
    CrearCilindro(5, 1.0f);//índice 3 en MeshList
    CrearCono(4, 1.5f); //índice 4 en MeshList
    CreateShaders();
}
```

Después comencé con la creación de un cubo rojo, un cono azul, un par de cubos verdes (los cuales servirán como puerta y ventanas), una esfera azul (la cual sirve como ventana trasera), un par de pirámides triangulares verdes (para los árboles) y finalmente otro par de cubos café (tronco del árbol).

```
//la línea de proyección solo se manda una vez a menos que en tiempo de ejecución
//se programe cambio entre proyección ortogonal y perspectiva
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular
```

```
//ejercicio: Instanciar primitivas geométricas para recrear el dibujo de la práctica pasada en 3D,
//se requiere que exista piso y la casa tiene una ventana azul circular justo en medio de la pared trasera y solo 1 puerta
model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 0.0f, 1.0f);
//Opcional duplicar esta traslación inicial para posicionar en -Z a los objetos en el mismo punto
//model = glm::translate(model, glm::vec3(0.0f, 0.0f, -3.0f));
model = glm::translate(model, glm::vec3(0.0f, 0.75f, -4.0f));
model = glm::scale(model, glm::vec3(0.45f, 0.45f, 0.45f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[4]->RenderMeshGeometry(); //creacion de cono azul

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 1.0f, 0.0f);
model = glm::translate(model, glm::vec3(0.0f, -0.35f, -3.5f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMeshGeometry(); //creacion de un cubo verde (puerta)

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 1.0f, 0.0f);
model = glm::translate(model, glm::vec3(-0.20f, 0.10f, -3.5f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMeshGeometry(); //creacion de un cubo verde (ventana izquierda)
```

```

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 1.0f, 0.0f);
model = glm::translate(model, glm::vec3(0.20f, 0.10f, -3.5f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMeshGeometry(); //creacion de un cubo verde (ventana derecha)

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 0.0f, 1.0f);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, -4.5f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
sp.render(); //creacion de un esfera azul (ventana trasera)

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 1.0f, 0.0f);
model = glm::translate(model, glm::vec3(0.80f, -0.03f, -3.94f));
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[1]->RenderMesh(); //creacion de piramide triangular verde (derecha)

```

```

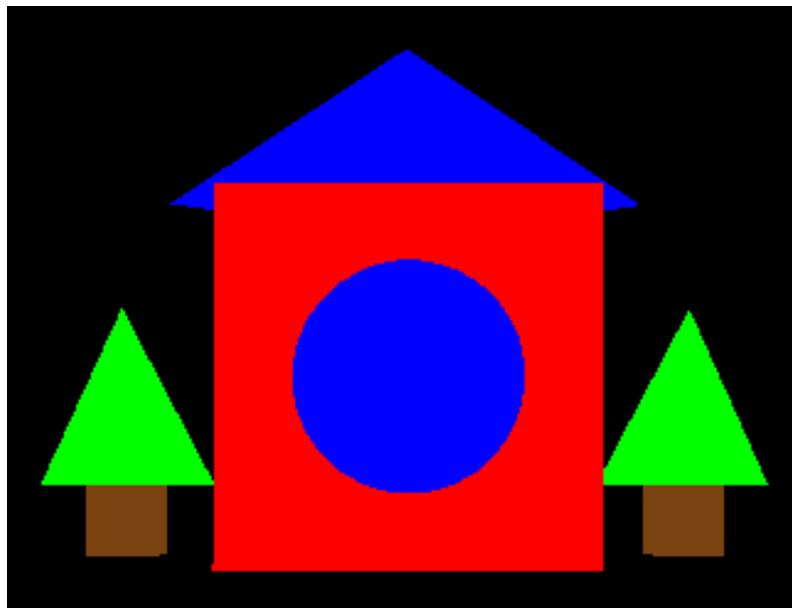
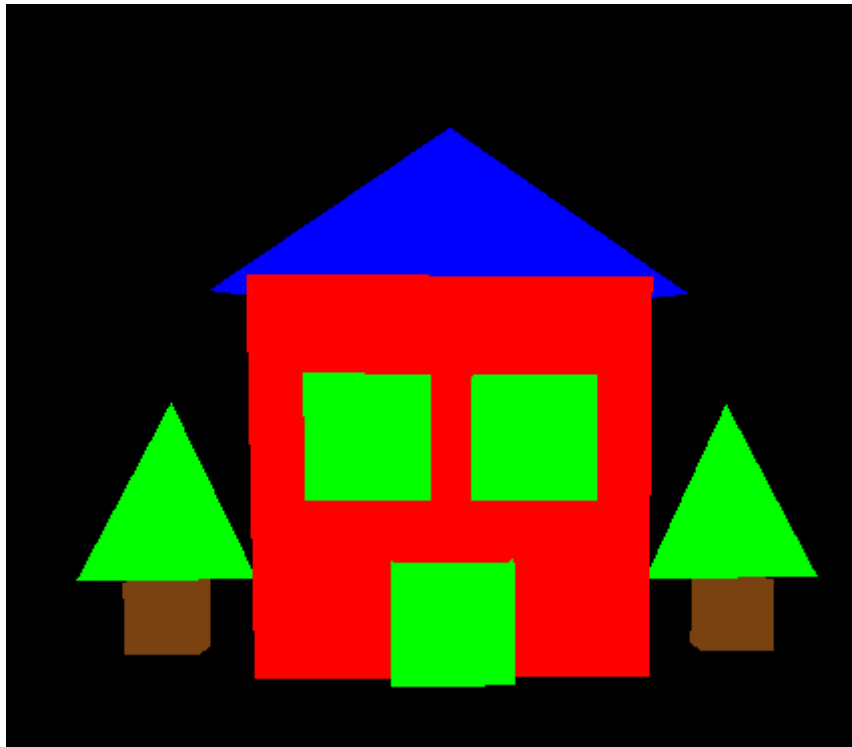
model = glm::mat4(1.0f);
color = glm::vec3(0.478f, 0.255f, 0.067f);
model = glm::translate(model, glm::vec3(0.80f, -0.38f, -3.94f));
model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMeshGeometry(); //creacion de cubo café (lado derecho)

model = glm::mat4(1.0f);
color = glm::vec3(0.0f, 1.0f, 0.0f);
model = glm::translate(model, glm::vec3(-0.80f, -0.03f, -3.94f));
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[1]->RenderMesh(); //creacion de piramide triangular verde (izquierda)

model = glm::mat4(1.0f);
color = glm::vec3(0.478f, 0.255f, 0.067f);
model = glm::translate(model, glm::vec3(-0.80f, -0.38f, -3.94f));
model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMeshGeometry(); //creacion de cubo café (lado izquierdo)

```

Ejecución:



2. Problemas presentados.

Los problemas que tuve fue un poco en la parte del techo de la casa ya que me confundí muchísimo con respecto a si debía elegir un cono o una pirámide triangular, finalmente me decanté por el cono y los problemas fueron tratar de acomodar el cono a la medida del cubo, como se observa en la imagen mi techo quedó un poco corto y considero que se puede mejorar.

3. Conclusiones

El ejercicio fue bastante interesante y resultó ser de muchísima paciencia, debido a que tuve que estar probando una y otra vez las coordenadas que se adaptaban más a lo solicitado y por ende fue bastante tedioso estar probando, aunque al final estoy conforme con el resultado que obtuve y me sentí más familiarizado con el lenguaje y la IDE.

La explicación por parte del profesor fue buena y gracias a la retroalimentación proporcionada, me fue muy útil ya que hubo una parte donde me perdí y gracias a la retro pude entender mejor lo que me solicitaban.