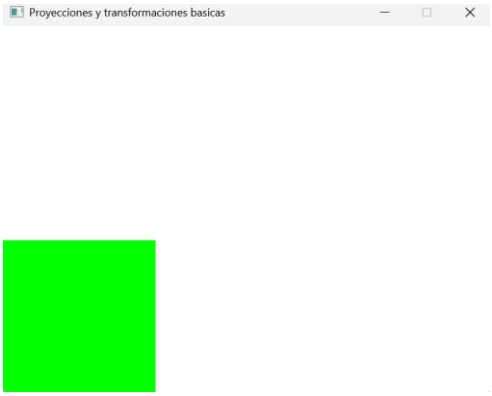
Chagoya González Leonardo
318218814

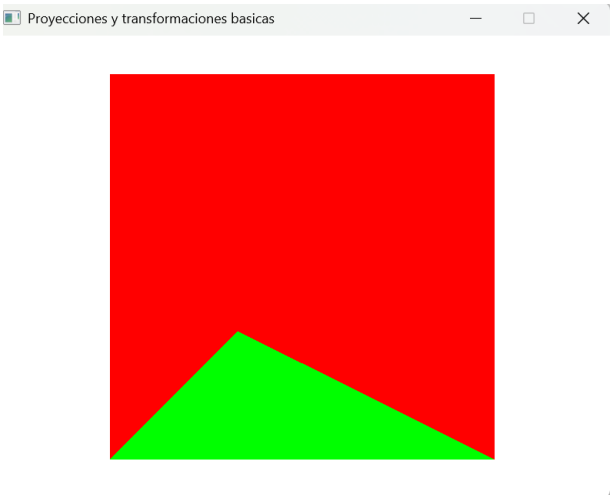
Proyección Ortogonal

Figura inicial



Visualizando dos caras del cubo en vista ortogonal

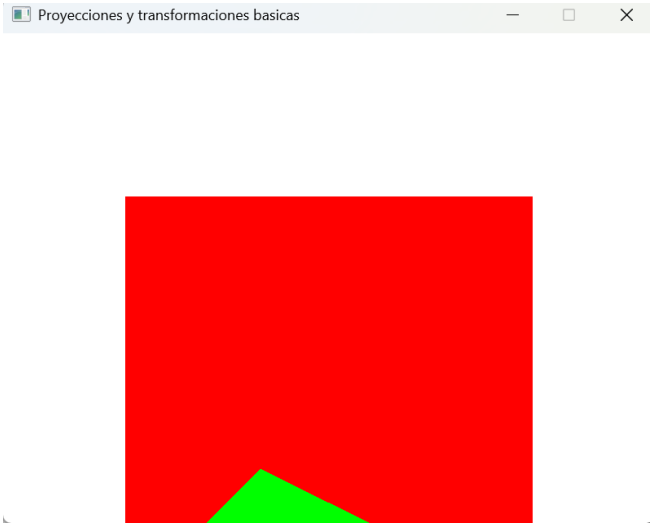
```
view = glm::translate( view, glm::vec3( screenWidth / 2, screenHeight / 2,-700.0f ) );
```



Trasladando la vista

```
view = glm::translate( view, glm::vec3( screenWidth / 2, screenHeight / 4,-700.0f ) );
```

Chagoya González Leonardo
318218814



Proyección en perspectiva

```
projection = glm::perspective(45.0f, (GLfloat)screenWidth /
(GLfloat)screenHeight, 0.1f, 100.0f);//FOV, Radio de aspecto,znear,zfar
```

Visualizando el elemento a una distancia muy cercana

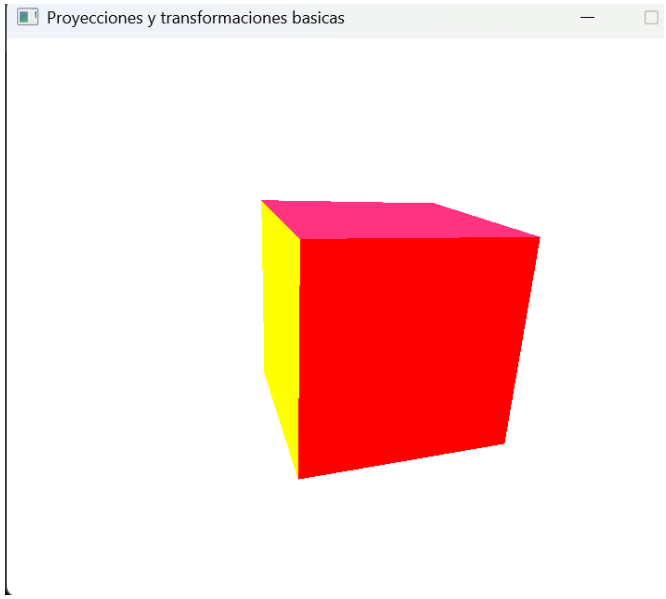
Alejando la vista

```
view = glm::translate(view, glm::vec3(0.0f,0.0f,-2.0f));
```

Chagoya González Leonardo
318218814

Rotando la figura en X y Y

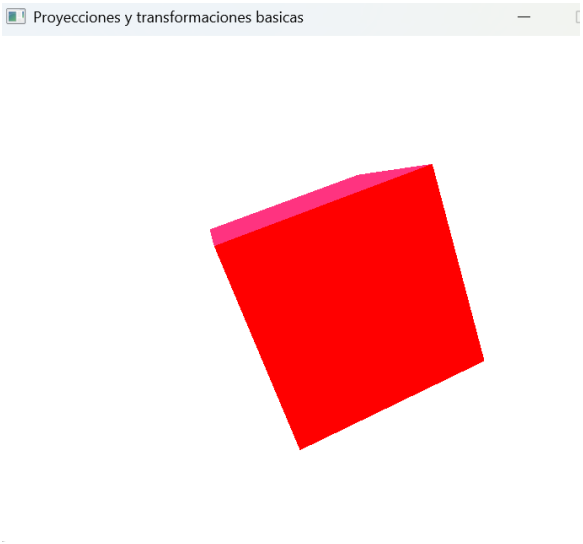
```
model = glm::rotate( model, 0.5f, glm::vec3( 2.0f, 2.0f, 0.0f ) );
```



Rotando la figura en X,Y.Z

```
model = glm::rotate( model, 0.5f, glm::vec3( 2.0f, 2.0f, 3.0f ) );
```

Chagoya González Leonardo
318218814



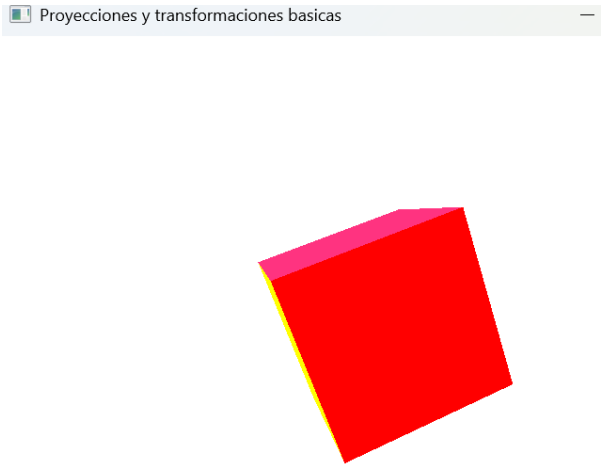
Escalando el cubo

```
model = glm::scale(model, glm::vec3(2.0f, 2.0f, 2.0f));
```

Escalando el elemento y alejando la vista

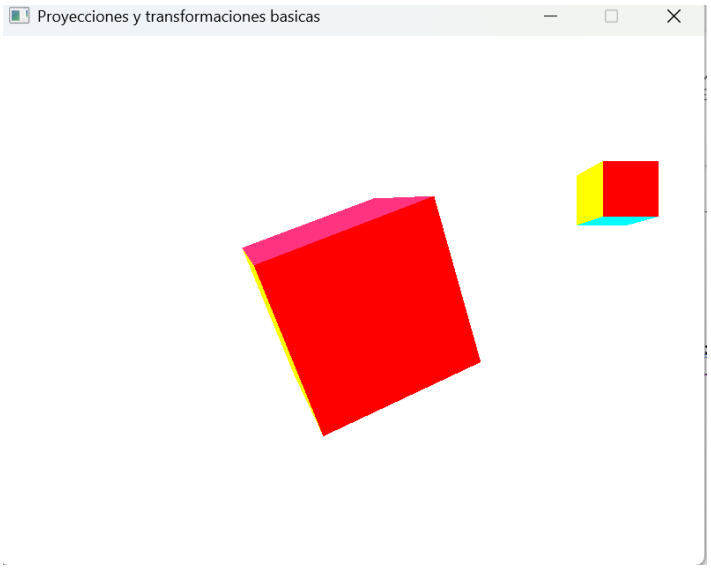
```
view = glm::translate(view, glm::vec3(0.0f,0.0f,-12.0f));model =
glm::rotate( model, 0.5f, glm::vec3( 0.0f, 1.0f, 0.0f ) ); // use to
compare orthographic and perspective projectionmodel = glm::scale(model,
glm::vec3(2.0f, 2.0f, 2.0f));
```

Chagoya González Leonardo
318218814



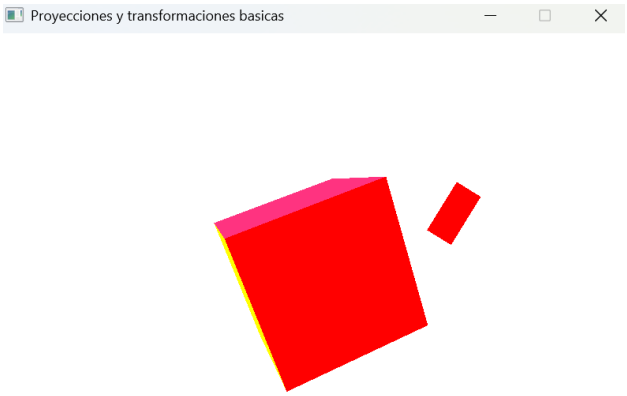
Creando un nuevo elemento

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(5.0f, 2.0f, 3.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
glBindVertexArray(0);
```



Aplicando la rotación y escalamiento al nuevo elemento

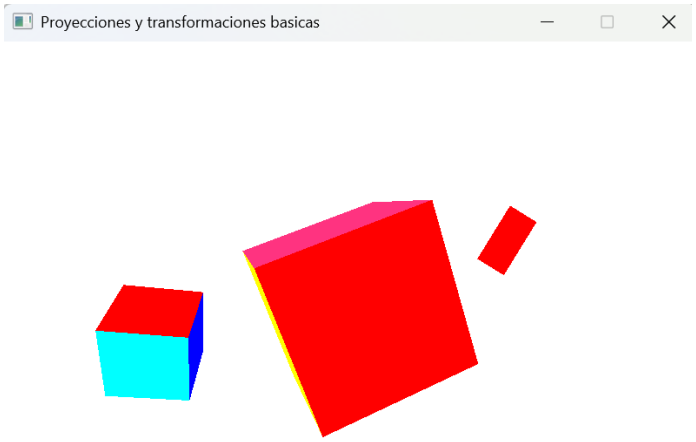
```
//creando un nuevo elemento
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(5.0f, 2.0f, -3.0f));
model = glm::rotate(model, 45.0f, glm::vec3(0.0f, 0.0f, 2.0f)); // use to
compare orthographic and perspective projection
model = glm::scale(model, glm::vec3(2.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
```



Creando un tercer elemento trasladándolo 5 puntos hacia la izquierda, 1 punto hacia abajo rotándolo en X,Y y escalándolo uniformemente en 2

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-5.0f, -1.0f, 0.0f));
model = glm::rotate(model, 45.0f, glm::vec3(-10.0f, 2.0f, 0.0f)); // use
to compare orthographic and perspective projection
model = glm::scale(model, glm::vec3(2.0f, 2.0f, 2.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
glDrawArrays(GL_TRIANGLES, 0, 36);
```

Chagoya González Leonardo
318218814



Bibliografía

Pérez, A. [Tecnologías Interactivas y Computación Gráfica] (16 de agosto de 2024). Proyecciones, Transformaciones y Shaders en OpenGL [Video]. YouTube: https://www.youtube.com/watch?v=2hnqyXRpURQ&list=PL9LBXPOWD3h3ZEc8z903fHSNKU5YyFeMI&index=5