

# Práctica de OpenGL e OpenSceneGraph

Nicolás Giraldez Amarelle      Pablo Rodríguez Pérez

21 de maio de 2025

## 1 OpenGL

Este ficheiro implementa unha escena 3D en OpenGL onde renderízanse un cubo, un cono e unha esfera, permitindo cambiar entre varias cámaras (fixas e orbital) mediante o teclado. O código utiliza GLFW para a xanela e eventos, GLEW para extensións de OpenGL e GLM para matemáticas de matrices e vectores.

### 1.1 Inicialización e *shaders*

- Inicialízanse GLFW y GLEW.
- Defínense *shaders* en GLSL:
  - *Vertex Shader*: Calcula a posición final do vértice e transforma as normais ao espazo mundial.
  - *Fragment Shader*: Implementa iluminación Phong (ambiental, difusa e especular) usando a posición da luz, a vista e a cor do obxecto.

### 1.2 Xeración de obxectos

- Cubo: Datos de vértices e normais definidos no vector `cubeVertices`.
- Cono: Xerado dinamicamente coa función `generateCone`, que calcula vértices e normais.
- Esfera: Xerada coa función `generateSphere`, que crea unha malla de triángulos.

Cada obxecto súbese á GPU usando VAO e VBO mediante as funcións `createObject` e `createObjectVec`.

### 1.3 Cámaras

Defínense varias cámaras no vector `cameras`, cada unha con posición, obxectivo (`target`) e vector up. Unha cámara orbital (a da posición 1) permite rotar arredor do cubo usando as frechas esquerda/dereita. O *callback* `key_callback` permite cambiar entre cámaras (1, 2, 3) ou rotar a cámara orbital.

## 1.4 Bucle de renderizado

En cada iteración:

1. Límpase a pantalla e o búfer de profundidade.
2. Calcúlase a vista e proxección segundo a cámara activa.
3. Envíanse as matrices e parámetros de iluminación aos *shaders*.
4. Debúxanse:

**Cubo:** Rótase e coloréase cada cara cunha cor diferente.

**Cono:** Debúxase á dereita e coloréase de verde.

**Esfera:** Debúxase á esquerda e coloréase de laranxa.

## 1.5 Controis

- Teclas 1, 2, 3: Cambian entre cámaras predefinidas.
- Frechas esquerda/dereita: Rotan a cámara orbital (cando está activa).

## 1.6 Limpeza

Ao final, elimínanse os VAO, VBO e o *shader program* para liberar recursos.

# 2 OpenSceneGraph

Toda a escena depende dun `osgViewer::Viewer`, do que depende o grupo que contén os obxectos e a luz. Os obxectos están formados por unha xerarquía de tres obxectos:

1. `osg::Geode`
2. `osg::Geometry`
3. `osg::Drawable`
4. `osg::Shape`

As cores múdanse na xeometría. A luz funciona dun xeito semellante, pero cunha xerarquía `osg::PositionAttitudeTransform` e `osg::LightSource`. `osg::LightSource` contén a luz, e `osg::PositionAttitudeTransform` indica a posición da fonte de luz.

Para as animacións créase un `osg::MatrixTransform`, este vincúlase a un *callback* e a animación empézase na función `main`. Aí tamén indícanse os fotogramas clave.

Para as cámaras creouse un xestor de eventos, que segundo a tecla pulsada muda a matriz da vista da cámara e toma unha fotografía.

### 3 Comparación

Mientras que OpenGL permite un control maior dos gráficos, OpenSceneGraph céntrase en xestionar os obxectos da escena. Isto permite empregar métodos de máis alto nivel. Por exemplo, o que en OpenSceneGraph é unha liña:

```
1 new osg::Box(osg::Vec3(0.0f,0.0f,0.0f),0.5)
```

en OpenGL son máis de 40:

```
1 float cubeVertices[] = {
2     // Positions      // Normals
3     -0.5f, -0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
4     0.5f, -0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
5     0.5f,  0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
6     0.5f,  0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
7     -0.5f,  0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
8     -0.5f, -0.5f, -0.5f,  0.0f, 0.0f, -1.0f,
9     [...]\n10     -0.5f,  0.5f, -0.5f,  0.0f, 1.0f, 0.0f,\n11 };
```

Isto é só a lista de puntos; despois inda hai que procesalos para que amosen un cubo. E este cubo é estático, a ollos de OpenGL desaparece tras crearse en pantalla. Polo tanto, se se quere mudar hai que volver a crealo modificando os puntos directamente. OpenSceneGraph permite unha manipulación directa dos parámetros do cubo.

Tamén coma resultado do anterior, OpenSceneGraph permite construír a árbore da escena, que aparte de facilitar a creación de obxectos, permite modificalos mudando só o obxecto pai e deixando que o resto muden en fervenza.

## 4 Capturas

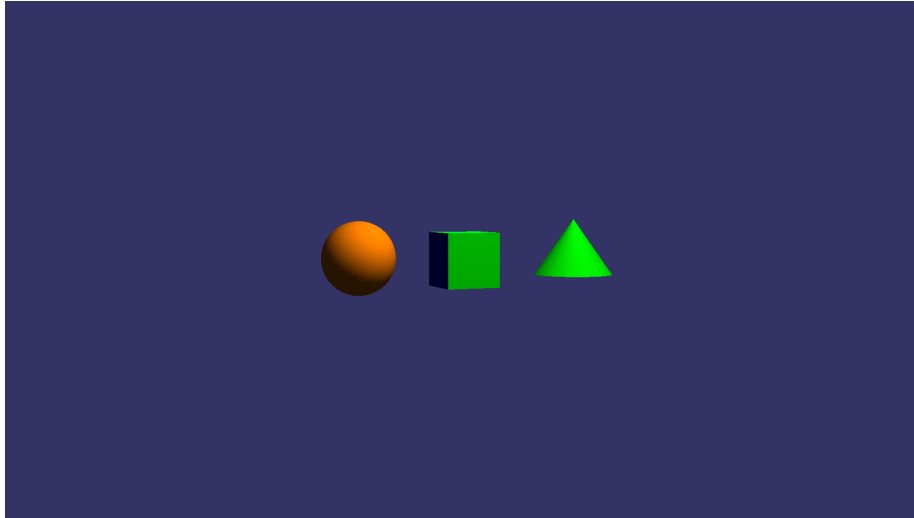


Figura 1: Cámara 1 de OpenSceneGraph

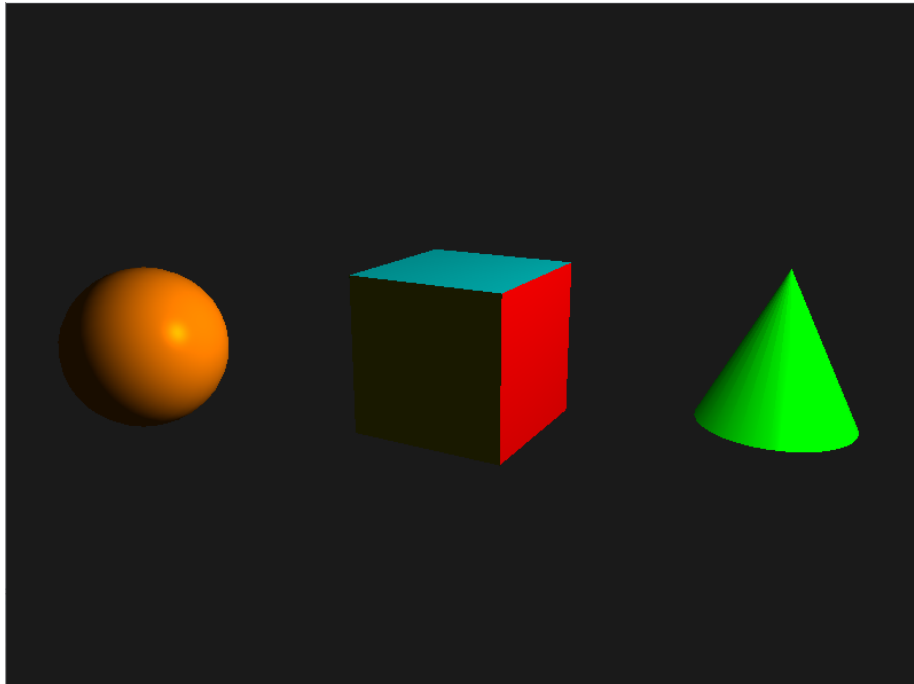


Figura 2: Cámara 1 de OpenGL

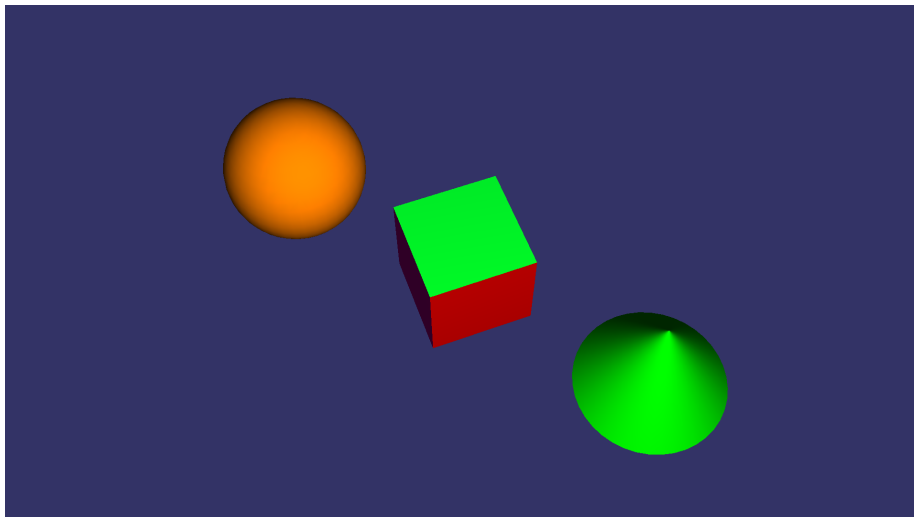


Figura 3: Cámara 2 de OpenSceneGraph

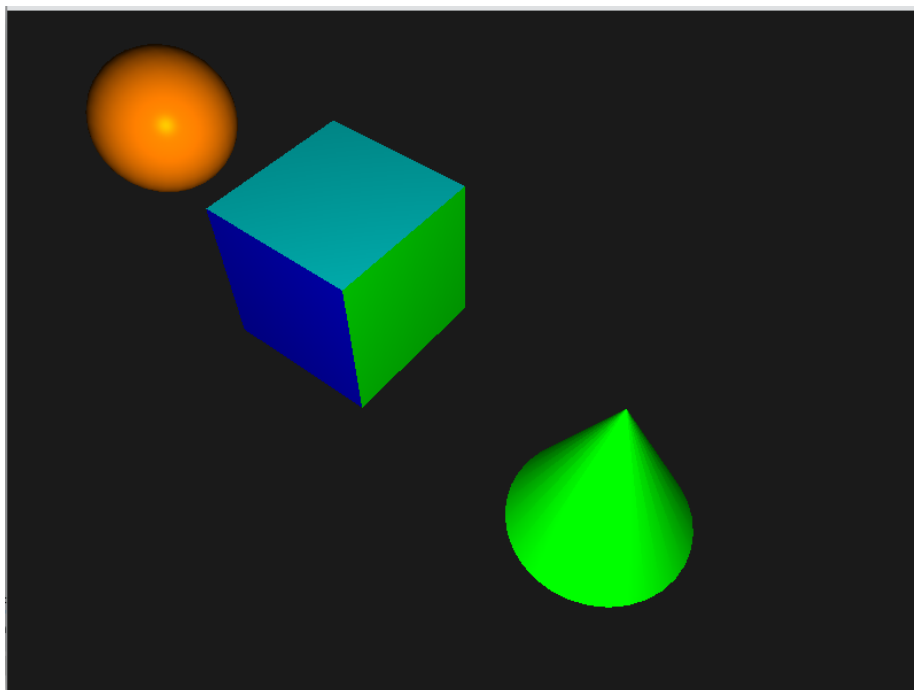


Figura 4: Cámara 2 de OpenGL

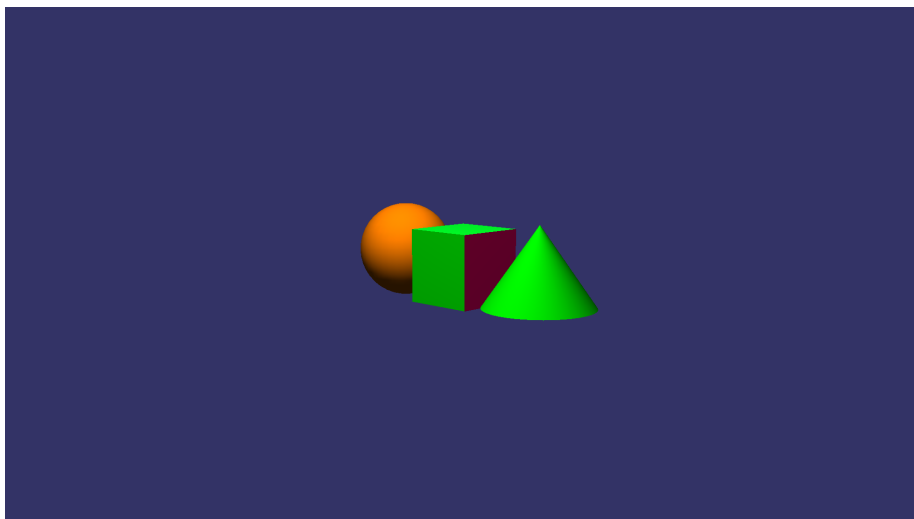


Figura 5: Cámara 3 de OpenSceneGraph

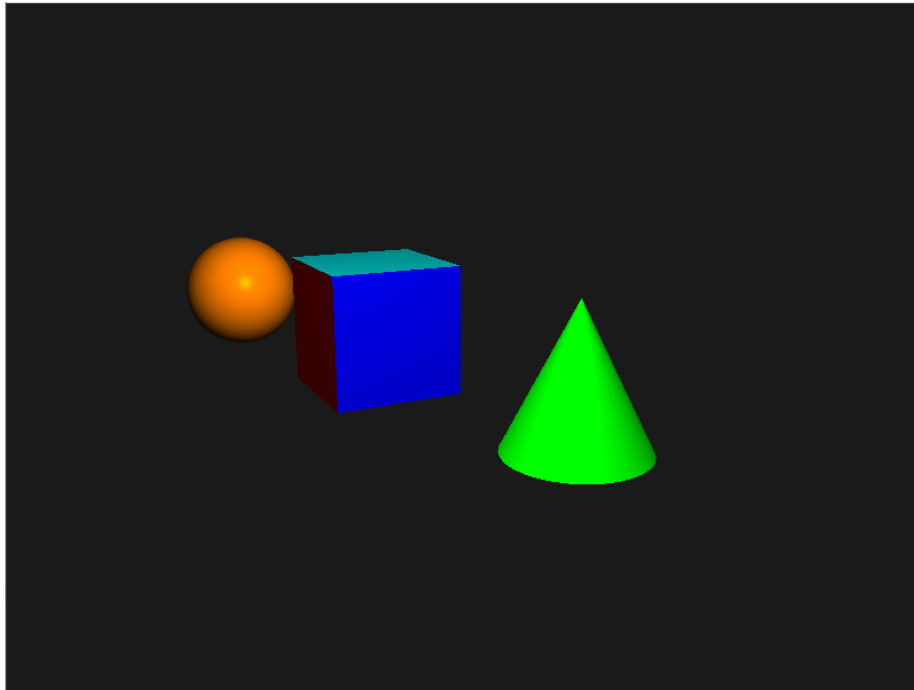


Figura 6: Cámara 3 de OpenGL