

Computer Graphics and Visualization

Practice N° 0

Introduction to OpenGL

(Open Graphics Library)

Practice objectives:

- Configuration of the development environment to be used in the lab
- Introduction to the basic concepts of the OpenGL graphics library:
 - basic template for an OpenGL program
 - the visualization process
 - display of basic primitives
 - keyboard event handling

Number of lab sessions to complete the practice: **one session (2 hours)**.

Score: **Introductory practice with no assignment or assigned grade**

Explanation:

All content necessary for completing the practical training, both theoretical and practical, must have been prepared and studied prior to the practical training. This will allow for maximum use of the practical training hours for programming and resolving doubts.

The practical sessions for this course, held in the practical classroom, will be carried out using the following software resources:

- C++ programming language
- OpenGL graphics library
- CLion programming environment

Development environment configuration:

Follow the instructions provided on the virtual teaching platform in the corresponding document.

Development:

This exercise explains how a basic OpenGL application works. It describes the **igvInterfaz** class and proposes exercises to help students understand basic concepts. Students should try to solve the exercises, the solutions to which will ultimately be provided and explained by the teaching staff.

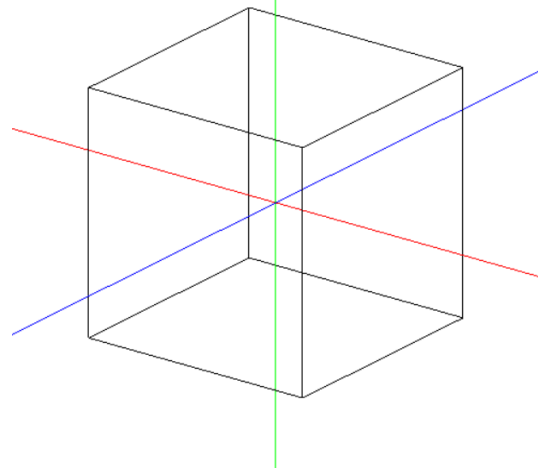
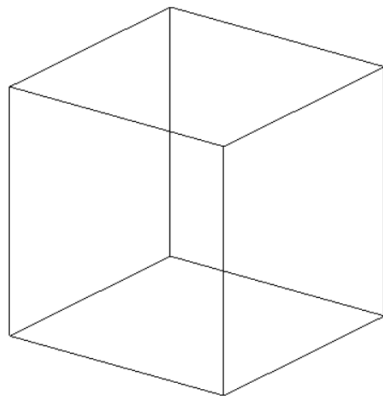
To carry out the exercise, we start with the code:

- **pr1.cpp**: main() function of the program
- **igvInterfaz.h** and **igvInterfaz.cpp**: Specification and implementation of the **igvInterfaz** class, which contains the basic functionality for creating a display window, its configuration, and the management of system events.

If we run the program without making any modifications, a window opens without any display, waiting for events. The following exercises are suggested:

- A) Make the necessary modifications so that the coordinate axes are displayed:
- Add the necessary code to the **igvInterfaz::displayFunc()** method so that the three axes are displayed, represented as three lines, centered at the origin and 200 units long. Assign the color red to the X-axis, the color green to the Y-axis, and the color blue to the Z-axis.
- B) Make the necessary modifications so that a cube is displayed in wireframe mode:
- Añade el código necesario en el método **igvInterfaz::displayFunc()** para que se visualice un cubo en modo alambre de color negro. El cubo debe estar centrado en el origen y tener de tamaño de lado una unidad. Utiliza la función: **void glutWireCube(GLdouble tamaño)**.
- C) Make the necessary modifications so that the display of the coordinate axes is activated/deactivated by pressing the 'e' key:
- Defines the Boolean attribute **igvInterface::axes** in the specification of the **igvInterface** class and its corresponding methods **igvInterface::get_axes()** and **igvInterface::set_axes()**. This attribute indicates whether the axes are enabled/disabled and must be initialized to true in the implementation of that class.
 - Add the necessary code to the **igvInterfaz::keyboardFunc()** method so that when the 'e' key is pressed, the display of the axes is activated/deactivated.

The final display that should be obtained is shown in the following figures (before and after pressing the 'e' key):



D) Add the necessary code to the `igvInterfaz::displayFunc()` method so that the top face of the cube in section B is displayed with two triangles of different colors, as shown in the figure (the colors do not need to match those shown in the figure):

