

Práctica 1: Dibujo de objetos geométricos. Proyecciones.

1 Objetivos

En esta práctica se pretenden como objetivos principales que el alumno aprenda a:

- representar objetos geométricos simples mediante la utilización de primitivas gráficas.
- manejo del buffer de color
- manejar el volumen de visualización.

Para ello se recomienda la utilización del manual docente *Introducción a OpenGL*, disponible en la página web de la asignatura.

2 Representación de objetos geométricos sencillos.

En este primer apartado, el alumno deberá componer la escena mostrada en la figura 1. Para ello deberá crear un programa donde se realicen los siguientes pasos:

- Establecer el color de fondo a azul.
- Dibujar un cubo verde en modo alambre.
- Dibujar triángulo rojo dentro del cubo en modo relleno.

El programa deberá ser nombrado como **practica1a**

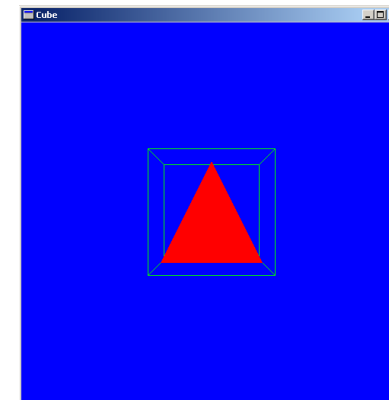


Figura 1.

3 Manejo del volumen de visualización.

Ejecutar el siguiente fragmento de código sustituyendo la llamada que aparece a `glFrustrum` con las distintas llamadas a `glFrustrum` y `gluPerspective` que se encuentran en comentarios. **Explicar ambas funciones y buscar una explicación a lo que sucede en cada caso.**

```
#include <windows.h>
#include <GL/glut.h>

void funcionDePintado (void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f (1.0, 1.0, 1.0);
    glLoadIdentity();
    glRotatef(45.0, 0.0, 1.0, 0.0);
    glTranslatef (5.0, 0.0, -5.0);
    glutWireSphere(2,10,10);
    glFlush();
}

void inicializar(){
    glClearColor(0, 0, 0, 1.0);
    glColor3f(0.0, 0.0, 1.0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

void funcionDeReescalado(GLsizei w, GLsizei h)
{
    glViewport(0, 0, w, h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    /*****
glFrustrum(-1.0, 1.0, -1.0, 1.0, 2.0, 10.0);
//glFrustrum(-1.0, 1.0, -1.0, 1.0, 5.0, 10.0);
//glFrustrum(-1.0, 1.0, -1.0, 1.0, 1.0, 10.0);
//gluPerspective(90.0, (float)w/(float)h, 2.0, 7.0);
//gluPerspective(30.0, (float)w/(float)h, 2.0, 7.0);
// gluPerspective(30.0, (float)w/(float)h, 2.0, 5.0);
//glFrustrum(-1.0, 1.0, -1.0, 1.0, 1.0, 5.5);
*****/
    glMatrixMode (GL_MODELVIEW); /* restaurar la matriz de modelo-vista como activa*/
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(250,250);
    glutInitWindowPosition(0,0);
    glutCreateWindow("Esfera");
    glutReshapeFunc(funcionDeReescalado);
    inicializar();
    glutDisplayFunc(funcionDePintado);
    glutMainLoop();
    return 0;
}
```