Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Computación Gráfica

“Ejercicio de Práctica 4”

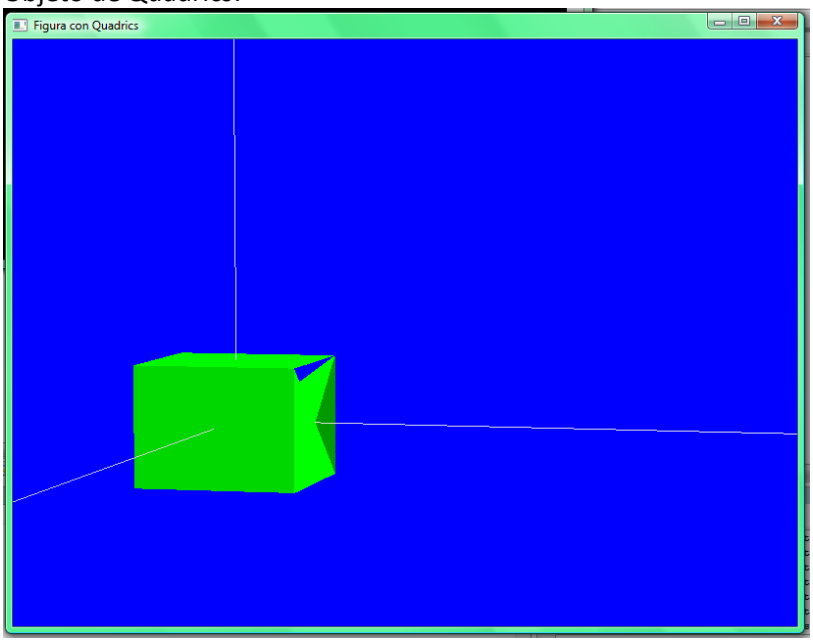
Profesora: MI Elizabeth Fonseca Chávez

Alumnos:
Magos Acuña Brenda
Pérez Cruz Juan Carlos

Grupo: 04

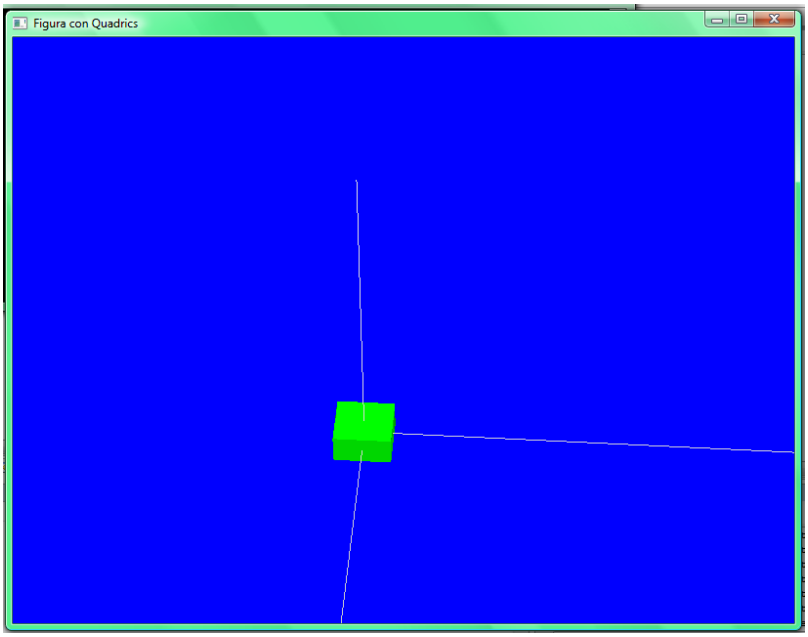
Miércoles 11 de Marzo de 2009

Objeto de Quadrics:

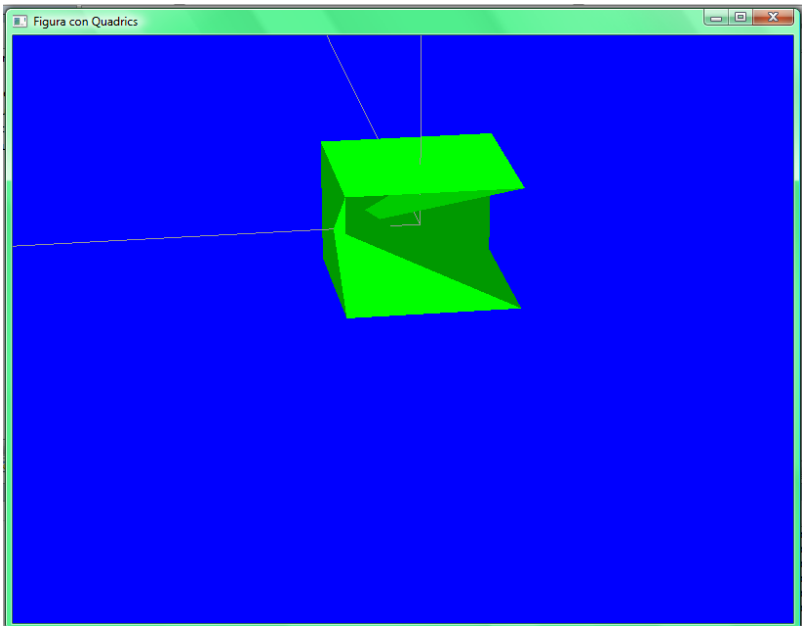


2 Vistas:

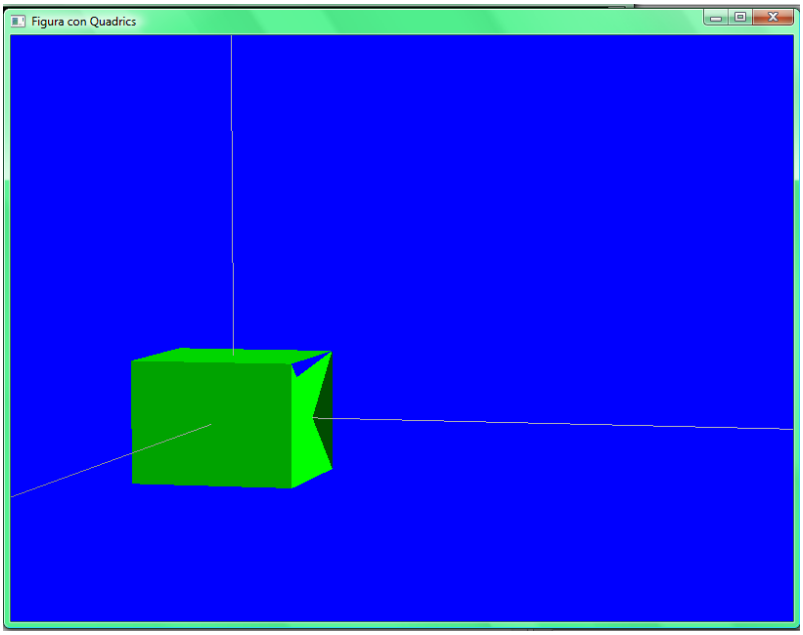
Vista de Arriba



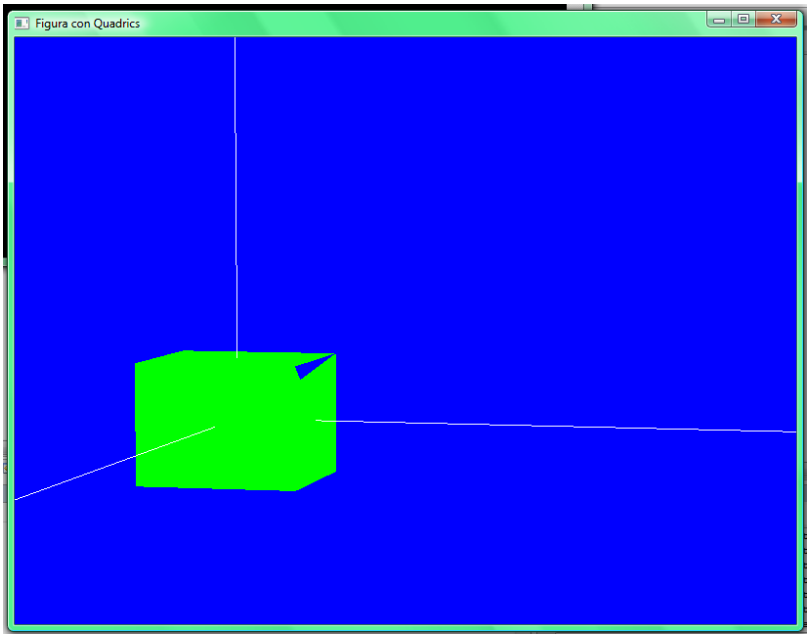
Vista desde atrás



Obscuro



Claro



Código

```
#include <gl/openglut.h>

//GLfloat light_Ambient  [4] = { 0.9,  0.9,  0.9, 1.0};//CLARO
//GLfloat light_Diffuse  [4] = { 0.5,  0.5,  0.5, 1.0};

//GLfloat light_Ambient  [4] = { 0.1,  0.1,  0.1, 1.0};//OBSCURO
//GLfloat light_Diffuse  [4] = { 1.0,  1.0,  1.0, 1.0};

GLfloat light_Ambient  [4] = { 0.4,  0.4,  0.4, 1.0};//ORIGINAL
GLfloat light_Diffuse  [4] = { 0.7,  0.7,  0.7, 1.0};
GLfloat light_Position [4] = {20.0, 15.0, 10.0, 1.0};

GLfloat material      [4] = {1.0, 0.2, 0.2, 1.0 };
GLfloat RedMaterial   [4] = {1.0, 0.0, 0.0, 1.0 };
GLfloat GreenMaterial [4] = {0.0, 1.0, 0.0, 1.0 };
GLfloat BlueMaterial  [4] = {0.0, 0.0, 1.0, 1.0 };
GLfloat WhiteMaterial [4] = {1.0, 1.0, 1.0, 1.0 };



void luces(void)
{

      glEnable (GL_LIGHTING);
      glEnable (GL_LIGHT0);

    glLightfv(GL_LIGHT0, GL_AMBIENT,  light_Ambient );
      glLightfv(GL_LIGHT0, GL_DIFFUSE,  light_Diffuse );
      glLightfv(GL_LIGHT0, GL_POSITION, light_Position );
}
```

```
void EjesReferencia()
{
  glNewList(1,GL_COMPILE);

  glBegin (GL_LINES);
      glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, WhiteMaterial );
        glVertex3f ( 0.0, 0.0, 0.0);
        glVertex3f (20.0, 0.0, 0.0);

      glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, WhiteMaterial );
        glVertex3f ( 0.0, 0.0, 0.0);
        glVertex3f ( 0.0,20.0, 0.0);

      glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, WhiteMaterial );
        glVertex3f ( 0.0, 0.0, 0.0);
        glVertex3f ( 0.0, 0.0,20.0);
   glEnd();

   glEndList();
}


void cubo()
{

      glNewList(2,GL_COMPILE);

      glMaterialfv  (GL_FRONT, GL_AMBIENT_AND_DIFFUSE, GreenMaterial );
      glPolygonMode (GL_FRONT_AND_BACK, GL_FILL);
      glShadeModel  (GL_FLAT);

    glBegin (GL_QUADS);
        glNormal3f ( 0.0, 0.0, 1.0);   //Frontal
        glVertex3f ( 1.0, 1.0, 1.0);
        glVertex3f (-1.0, 1.0, 1.0);
        glVertex3f (-1.0,-1.0, 1.0);
        glVertex3f ( 1.0,-1.0, 1.0);
        glVertex3f ( 1.0,0.0, 0.0);

        glNormal3f ( 0.0, 0.0,-1.0);   //Posterior
        glVertex3f ( 1.0, 1.0,-1.0);
        glVertex3f ( 1.0,-1.0,-1.0);
        glVertex3f (-1.0,-1.0,-1.0);
        glVertex3f (-1.0, 1.0,-1.0);
        glVertex3f ( 1.0,0.0, 0.0);

        glNormal3f (-1.0, 0.0, 0.0);  //Izquierda
        glVertex3f (-1.0, 1.0, 1.0);
        glVertex3f (-1.0, 1.0,-1.0);
        glVertex3f (-1.0,-1.0,-1.0);
        glVertex3f (-1.0,-1.0, 1.0);
        glVertex3f ( 1.0,0.0, 0.0);

        glNormal3f ( 1.0, 0.0, 0.0);  //Derecha
        glVertex3f ( 1.0, 1.0,-1.0);
        glVertex3f ( 1.0,-1.0,-1.0);
```

```
        glVertex3f ( 1.0,-1.0, 1.0);
        glVertex3f ( 1.0, 1.0, 1.0);
        glVertex3f ( 1.0,0.0, 0.0);

        glNormal3f ( 0.0, 1.0, 0.0);  //Arriba
        glVertex3f (-1.0, 1.0,-1.0);
        glVertex3f (-1.0, 1.0, 1.0);
        glVertex3f ( 1.0, 1.0, 1.0);
        glVertex3f ( 1.0, 1.0,-1.0);
        glVertex3f ( 0.0,-1.0, 1.0);

        glNormal3f ( 0.0,-1.0, 0.0);  //Abajo
        glVertex3f (-1.0,-1.0,-1.0);
        glVertex3f ( 1.0,-1.0,-1.0);
        glVertex3f ( 1.0,-1.0, 1.0);
        glVertex3f (-1.0,-1.0, 1.0);
        glVertex3f ( 1.0,0.0, 0.0);

    glEnd();

      glEndList();
}

void RenderScene(void)
{

      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
      glMatrixMode(GL_MODELVIEW);
      glLoadIdentity();
      //gluLookAt(5,90,50,1,1,-5,0,1,0);//Vista desde arriba
      gluLookAt(5,15,-30,1,1,-5,0,1,0);//Vista desde atras
      glCallList(1);
      glCallList(2);

      glFlush();
      }

int main(void)
{
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |  GLUT_DEPTH);
      glutInitWindowSize(800,600);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Figura con Quadrics");
    glutDisplayFunc(RenderScene);
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
    glEnable(GL_DEPTH_TEST);
     glMatrixMode(GL_PROJECTION);
       glLoadIdentity();
       gluPerspective(15,800/600,.1,1000);
       luces();
      EjesReferencia();
    cubo();

glutMainLoop();
      return 0;
}
```