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


static GLfloat light0_position[] = {0.0, 2.0, 2.0, 0.0};
static GLfloat light1_position[] = {-2.0, -2.0, -2.0, 0.0};

GLfloat mat_specular[]         = {1.0, 1.0, 1.0, 1.0};
GLfloat mat_amb_diff_white[]   = {1.0, 1.0, 1.0, 1.0};
GLfloat mat_amb_diff_green[]   = {0.0, 1.0, 0.0, 1.0};
GLfloat mat_amb_diff_blue[]    = {0.0, 0.0, 1.0, 1.0};
GLfloat mat_amb_diff_cyan[]    = {0.0, 1.0, 1.0, 1.0};
GLfloat mat_amb_diff_red[]     = {1.0,0.0,0.0,1.0};
float spin=0.0;
float v=1.0;
int shape='t';

void init()
{
  glClearColor(0.0, 0.0, 0.0, 0.0);
  glShadeModel(GL_SMOOTH);
  glEnable(GL_DEPTH_TEST);
  glEnable(GL_LIGHTING);
  glEnable(GL_LIGHT0);
  glEnable(GL_LIGHT1);
}

void drawaxes()
{
  glLineWidth(2.0);
  glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_amb_diff_white);
  glBegin(GL_LINES);
    glVertex3f(-7,0,0);
    glVertex3f(7,0,0);
    glVertex3f(0,-7,0);
    glVertex3f(0,7,0);
    glVertex3f(0, 0, -7);
    glVertex3f(0, 0, 7);
  glEnd();
}

void drawcube()
{
        glLineWidth(4.0);
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE,mat_amb_diff_cyan );
        //Sxedio mpros pleuras
        glBegin(GL_LINE_LOOP);
                glVertex3f(2,2,2);
                glVertex3f(-2,2,2);
                glVertex3f(-2,-2,2);
                glVertex3f(2,-2,2);
        glEnd();
        //Sxedio pisw pleuras
        glBegin(GL_LINE_LOOP);
                glVertex3f(2,2,-2);
                glVertex3f(-2,2,-2);
                glVertex3f(-2,-2,-2);
                glVertex3f(2,-2,-2);
        glEnd();
        //sxedio endiameswn grammwn
        //proseksete thn diafora apo GL_LINE_LOOP
        //se GL_LINES dwste:
```



```
        //man glBegin gia na diabasete tis diafores.
        glBegin(GL_LINES);
                glVertex3f(2,2,2);
                glVertex3f(2,2,-2);
                glVertex3f(-2,2,2);
                glVertex3f(-2,2,-2);
                glVertex3f(-2,-2,2);
                glVertex3f(-2,-2,-2);
                glVertex3f(2,-2,2);
                glVertex3f(2,-2,-2);
        glEnd();


}

void drawlocalaxes()
{
  glLineWidth(2.0);
  glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_amb_diff_cyan);
  glBegin(GL_LINES);
    glVertex3f(-3,0,0);
    glVertex3f(3,0,0);
    glVertex3f(0,-3,0);
    glVertex3f(0,3,0);
    glVertex3f(0, 0, -3);
    glVertex3f(0, 0, 3);
  glEnd();
}

void reshape (int w, int h)
{
  glViewport(0,0, (GLsizei) w, (GLsizei) h);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(45.0, (GLfloat) w / (GLfloat) h, 2.0, 20.0);
  glMatrixMode(GL_MODELVIEW);
}

void display()
{
  glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
  /* Clear the matrix */
  glLoadIdentity();
  gluLookAt(5, 5, 10, 0 , 0 , 0, 0, 1, 0);
  glLightfv(GL_LIGHT0, GL_POSITION, light0_position);
  glLightfv(GL_LIGHT1, GL_POSITION, light0_position);


  drawaxes();
  glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_amb_diff_green);


  spin=spin+v;
  if (spin > 360.0)
     spin = spin - 360.0;

  glPushMatrix();
  glRotatef(spin,0.0,0.0,1.0);

  drawcube();
  glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_amb_diff_red);
```

```
  drawEswteriko();

  printf("%f\n",v);
  glPopMatrix();
  glutSwapBuffers();
}


void
key(unsigned char key, int x, int y)
{
  menu((int) key);
}

void drawEswteriko()
{
        switch (shape){

        case 's':
                        glutSolidSphere(1.0,20,20);
        break;

        case 'T':
                        glutSolidTorus(0.5,1.2,20,20);
        break;

        case 't':
                        glutSolidTeapot(1.0);
        break;

        case 'i':
                        glutSolidIcosahedron();
        break;

        case 'k':
                        glutSolidCone(1.0,1.0,20,20);
        break;

        }
}

void menu(int item)
{
        switch (item) {
                case 'F':
                        v=v+0.5;
        break;
        case 'f':
                        v=v-0.5;
        break;
        case 's':
                        shape='s';
        break;

        case 'T':
                        shape='T';
        break;

        case 't':
                        shape='t';
        break;
```



```
        case 'i':
                        shape='i';

        break;
        case 'k':
                        shape='k';
        break;

        case '\033':            /* ESC key: quit */
          exit(0);
        break;

        }
}









int main(int argc, char** argv)
{
  glutInit(&argc, argv);
  /*
   * Prosekste to GL_DOUBLE sthn epomenhn grammh.
   * Energopoiei to double buffering to opoio exei
   * san apotelesma omaloterh kinhsh.
   * Dokimaste na allaksete thn parapanw grammh se
   * single buffering gia na deite thn diafora.
   */
  glutInitDisplayMode(GLUT_DOUBLE |GLUT_RGB);
  glutInitWindowSize(350, 350);
  glutInitWindowPosition(0, 0);
  glutCreateWindow("Cube Rotation");


  /*
   *     Oi parakatw entoles ka8orizoun to menu
   *     Mporoume na allaksoume ta sxhmata peristrefoume
   */

   glutCreateMenu(menu);

   glutAddMenuEntry("F: Aukshsh Taxythtas Peristrofhs", 'F');
   glutAddMenuEntry("f: Meiwsh Taxythtas Peristrofhw", 'f');
   glutAddMenuEntry("s: Allagh se sfaira", 's');
   glutAddMenuEntry("t: Allagh se tsagiera", 't');
   glutAddMenuEntry("T: Allagh se toro", 'T');
   glutAddMenuEntry("k: Allagh se kono", 'k');
   glutAddMenuEntry("i: Allagh se Ikosaedro", 'i');
   glutAddMenuEntry("<esc>: exit program", '\033');

   glutAttachMenu(GLUT_RIGHT_BUTTON);

   glutKeyboardFunc(key);
```

```
  init();
  glutDisplayFunc(display);

  // H parakatw synarthsh kaleite synexeia
  // gia na sxediasei
   glutIdleFunc(display);
//
//H parakatw synarthsh kaleite otan "trabame" to para8yro
//gia na to megalwsoume h’ gia na to mikrynoume
  glutReshapeFunc(reshape);
/*
 * To initialization teleiwse, seira ths OpenGL na
 * parei ta inia.
 */
  glutMainLoop();
  return 0;
}
```