

```

#include <GL/glut.h>
#include <stdlib.h>

//Declaracion de Funciones
void display(void);
void myInit (void);
void reshape(int w, int h);
void keyboard(unsigned char key, int x, int y);
void tecladoEspecial(int key, int x, int y);

//Variable global para controlar la luz
int hayLuz;

//Rotacion de la camara
float rotaX, rotaY;

void display(void){
    //Color del poligono
    float color[4] = {0.784313725, 0.509803922, 0.22745098, 1.0};

    //Material del poligono
    float mat_dela_ambiente[4] = {0.65, 0.75, 0.65, 1.0},
          mat_dela_difuso[4] = {0.75, 0.85, 0.75, 1.0},
          mat_dela_especular[4] = {0.90, 0.9, 0.9, 1.0},
          mat_tras_ambiente[4] = {0.25, 0.1, 0.1, 1.0},
          mat_tras_difuso[4] = {0.35, 0.1, 0.1, 1.0},
          mat_tras_especular[4] = {1.0, 1.0, 1.0, 1.0};

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    if(hayLuz) {
        glEnable(GL_LIGHTING);
        //Material del poligono de adelante
        glMaterialfv(GL_FRONT, GL_AMBIENT, mat_dela_ambiente);
        glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_dela_difuso);
        glMaterialfv(GL_FRONT, GL_SPECULAR, mat_dela_especular);
        glMaterialf(GL_FRONT, GL_SHININESS, 128);
        //Material del poligono de atras
        glMaterialfv(GL_BACK, GL_AMBIENT, mat_tras_ambiente);
        glMaterialfv(GL_BACK, GL_DIFFUSE, mat_tras_difuso);
        glMaterialfv(GL_BACK, GL_SPECULAR, mat_tras_especular);
        glMaterialf(GL_BACK, GL_SHININESS, 1);
    } else {
        glDisable(GL_LIGHTING);
        //Color del poligono
        glColor4fv(color);
    }

    glPushMatrix();
    glRotatef(rotaX, 1.0, 0.0, 0.0);
    glRotatef(rotaY, 0.0, 1.0, 0.0);

    glBegin(GL_POLYGON);
    glVertex3f(-3.0, -3.0, 0.0);
    glVertex3f(-1.5, -3.0, 0.0);
    glVertex3f(3.0, 3.0, 0.0);
    glVertex3f(1.5, 3.0, 0.0);
    glEnd();
    glPopMatrix();

    glutSwapBuffers();
}

void myInit () {
    float luz_ambiente[4] = {0.5, 0.5, 0.5, 1.0},
          luz_difusa[4] = {0.5, 0.5, 0.5, 1.0},
          luz_especular[4] = {0.0, 0.0, 0.0, 1.0};

    glClearColor(1.0, 1.0, 1.0, 1.0);

    glEnable(GL_LIGHT0);

```

```

    glEnable(GL_DEPTH_TEST);

    //De display
    rotaX = 0.0;
    rotaY = 0.0;
    hayLuz = 0;

    //Color de la luz
    glLightfv(GL_LIGHT0, GL_AMBIENT, luz_ambiente);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, luz_difusa);
    glLightfv(GL_LIGHT0, GL_SPECULAR, luz_especular);

    //Del modelo de la luz
    glLightModeli(GL_LIGHT_MODEL_TWO_SIDE, GL_TRUE);
}

void reshape(int w, int h){

    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5.0, 5.0, -5.0, 5.0, -5.0, 5.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

}

void keyboard(unsigned char key, int x, int y){

    switch (key) {
        case 'L':
        case 'l':
            hayLuz = !(hayLuz);
            break;

        case 27:
            exit(0);
            break;
    }

    glutPostRedisplay();
}

void tecladoEspecial(int key, int x, int y) {
    switch (key) {
        case GLUT_KEY_UP:
            rotaX -= 10.0;
            break;

        case GLUT_KEY_DOWN:
            rotaX += 10.0;
            break;

        case GLUT_KEY_LEFT:
            rotaY -= 10.0;
            break;

        case GLUT_KEY_RIGHT:
            rotaY += 10.0;
            break;
    }

    glutPostRedisplay();
}

int main(int argc, char** argv){

    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("Luz");
}

```

```
myInit();

glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutKeyboardFunc (keyboard);
glutSpecialFunc (tecladoEspecial);

glutMainLoop();

return 0;
}
```