

```

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

#define MUNDO 4.0f
float rotaY, rotaMuslo, rotaPantorrilla, rotaPie;

void display(void);
void keyboard (unsigned char key, int x, int y);
void reshape (int w, int h);
void inicializar(void);

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("Pie de un Robot");
    glEnable(GL_DEPTH_TEST);

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();

    return 0;
}

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix();
    glRotatef (rotaY, 0.0, 1.0, 0.0); //La rotacion de la camara

    glPushMatrix(); //Todo lo que conforma la pierna

    glRotatef (rotaMuslo, 0.0, 0.0, 1.0); //La rotacion del muslo
    //Dibuja el muslo
    glPushMatrix();
    glTranslatef (0.0, 1.6, 0.0);
    glScalef (0.6, 2.0, 0.5);
    glColor3f(1.0, 0.0, 1.0);
    glutSolidCube (1.0); //muslo
    glPopMatrix();
    //Dibuja la rodilla
    glPushMatrix();
    glTranslatef (0.0, 0.3, 0.0);
    glColor3f(0.7, 0.7, 0.7);
    glScalef(0.29, 0.29, 0.29);
    glutSolidSphere(1.0, 15, 15); //rodilla
    glPopMatrix();

    glPushMatrix();
    //La rotacion de la pantorrilla
    glRotatef (rotaPantorrilla, 0.0, 0.0, 1.0);
    //Dibuja la pantorrilla
    glPushMatrix();
    glTranslatef (0.0, -1.0, 0.0);
    glScalef (0.6, 2.0, 0.5);
    glColor3f(0.0, 0.0, 1.0);
    glutSolidCube (1.0); //pantorrilla
    glPopMatrix();
    //Dibuja el talon
    glPushMatrix();
    glTranslatef (0.0, -2.3, 0.0);
    glColor3f(0.7, 0.7, 0.7);
    glScalef(0.28, 0.28, 0.28);
    glutSolidSphere(1.0, 15, 15); //talon
    glPopMatrix();

    glPushMatrix();
    glTranslatef(0.0, -2.8, 0.0);

```

```

        glRotatef (rotaPie, 0.0, 0.0, 1.0); //La rotacion del pie
        glTranslatef (0.0, 2.8, 0.0);
        //Dibuja el pie
        glPushMatrix();
            glTranslatef (0.25, -2.8, 0.0);
            glScalef (1.0, 0.35, 0.7);
            glColor3f(0.5, 0.0, 0.8);
            glutSolidCube (1.0); //pie
        glPopMatrix();
        glPopMatrix(); //La rotacion del pie
        glPopMatrix(); //La rotacion de la pantorrilla
        glPopMatrix(); //La rotacion del muslo

    glPopMatrix(); //La de la camara

    glutSwapBuffers();
}
void keyboard (unsigned char key, int x, int y)
{
    switch (key) {
        case 27:
            exit(0);
            break;

        case 'M':
            rotaMuslo += 5.0;
            break;

        case 'm':
            rotaMuslo -= 5.0;
            break;

        case 'A':
            if (rotaPantorrilla >= 5.0)
                rotaPantorrilla = 5.0;
            else
                rotaPantorrilla += 5.0;
            break;

        case 'a':
            if(rotaPantorrilla <= -75.0 )
                rotaPantorrilla = -75.0;
            else
                rotaPantorrilla -= 5.0;
            break;

        case 'P':
            if (rotaPie >= 30.0)
                rotaPie = 30.0;
            else
                rotaPie += 5.0;
            break;

        case 'p':
            if (rotaPie <= -30.0)
                rotaPie = -30.0;
            else
                rotaPie -= 5.0;
            break;

        case 'r':
            rotaY += 2.0; //rota la camara con respecto a y
            break;

        case 'R':
            rotaY -= 2.0;
            break;

        default: break;
    }
}

glutPostRedisplay();

```

```

}

void reshape (int w, int h)
{
    float aspect;
    glViewport (0, 0, w, h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();

    if (h == 0){
        h = 1;
    }

    aspect = (float) w / (float) h;

    if (w <= h){
        glOrtho(-MUNDO, MUNDO, -MUNDO/aspect, MUNDO/aspect, -MUNDO, MUNDO);
    } else {
        glOrtho(-MUNDO * aspect, MUNDO * aspect, -MUNDO, MUNDO, -MUNDO, MUNDO);
    }

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void inicializar(void) {
    glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    glEnable(GL_DEPTH_TEST);
    rotaY = 0.0f;
    rotaMuslo = 0.0f;
    rotaPantorrilla = 0.0f;
    rotaPie = 0.0f;
}

```