UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Computación Gráfica

FECHA DE ENTREGA: 5 / *MARZO / 2019*

TORRES CABALLERO BRUNO

SEMESTRE 2019-2

Código:

```
//#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>
//función para dibujar un cubo a partir de polígonos

float transZ = -10.0f;
float transX = 0.0f;
float transY = 0.0f;

float angleX = 0.0f;
float angleY = 0.0f;

/*float cabello[3] = { 0.007,0.011,0.031 };
float traje[3] = { 0.788,0.792,0.792 };
float botas[3] = { 0.007,0.011,0.031 };
float capa[3] = { 0.015,0.058,0.286 };
float cinturon[3] = { 0.949,0.788,0.109 };
float piel[3] = { 0.925,0.756,0.396 };
*/

float cabello[3] = { 0.007,0.011,0.031 };
float traje[3] = { 0.149,0.360,0.890 };
float botas[3] = { 0.866,0.011,0.133 };
float capa[3] = { 0.866,0.011,0.133 };
float cinturon[3] = { 0.949,0.788,0.109 };
float piel[3] = { 0.925,0.756,0.396 };

void inicializar(void) {

    glClearColor(0.831f, 0.949f, 0.988f, .1f);
    glClearDepth(1.0f);
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
}

void prisma(void)
{
    GLfloat vertice [8][3] = {
                {0.5 ,-0.5, 0.5},    //Coordenadas Vértice 0 V0
                {-0.5 ,-0.5, 0.5},    //Coordenadas Vértice 1 V1
                {-0.5 ,-0.5, -0.5},    //Coordenadas Vértice 2 V2
                {0.5 ,-0.5, -0.5},    //Coordenadas Vértice 3 V3
                {0.5 ,0.5, 0.5},    //Coordenadas Vértice 4 V4
                {0.5 ,0.5, -0.5},    //Coordenadas Vértice 5 V5
                {-0.5 ,0.5, -0.5},    //Coordenadas Vértice 6 V6
                {-0.5 ,0.5, 0.5},    //Coordenadas Vértice 7 V7
                };

        glBegin(GL_POLYGON);    //Enfrente
            glColor3fv(traje);
            glVertex3fv(vertice[0]);
            glVertex3fv(vertice[4]);
            glVertex3fv(vertice[7]);
            glVertex3fv(vertice[1]);
        glEnd();
```

```
        glBegin(GL_POLYGON);    //Derecha
            glColor3fv(capa);
            glVertex3fv(vertice[0]);
            glVertex3fv(vertice[3]);
            glVertex3fv(vertice[5]);
            glVertex3fv(vertice[4]);
        glEnd();

        glBegin(GL_POLYGON);    //Atrás
            glColor3fv(cinturon);
            glVertex3fv(vertice[6]);
            glVertex3fv(vertice[5]);
            glVertex3fv(vertice[3]);
            glVertex3fv(vertice[2]);
        glEnd();

        glBegin(GL_POLYGON);  //Izquierda
            glColor3fv(botas);
            glVertex3fv(vertice[1]);
            glVertex3fv(vertice[7]);
            glVertex3fv(vertice[6]);
            glVertex3fv(vertice[2]);
        glEnd();

        glBegin(GL_POLYGON);  //Abajo
            glColor3fv(piel);
            glVertex3fv(vertice[0]);
            glVertex3fv(vertice[1]);
            glVertex3fv(vertice[2]);
            glVertex3fv(vertice[3]);
        glEnd();

        glBegin(GL_POLYGON);  //Arriba
            glColor3fv(cabello);
            glVertex3fv(vertice[4]);
            glVertex3fv(vertice[5]);
            glVertex3fv(vertice[6]);
            glVertex3fv(vertice[7]);
        glEnd();
}

void teclado(unsigned char key, int x, int y)
{
    switch ( key ) {
        case 'w': case 'W':  //acerca al objeto con traslación en z pos
            transZ +=0.2f;
            break;

        case 's': case 'S':  //aleja al objeto con traslación en z neg
            transZ -=0.2f;
            break;

        case 'a': case 'A':  //traslada objeto hacia la derecha en x pos
            transX +=0.2f;
            break;

        case 'd': case 'D':  //traslada objeto hacia la izquierda en x neg
            transX -=0.2f;
            break;
```

```
        case 'e': case 'E':  //traslada objeto hacia arriba en y pos
            transY +=0.2f;
            break;

        case 'c': case 'C':  //traslada objeto hacia abajo en y neg
            transY -=0.2f;
            break;

        case 27:  //Si presiona tecla ESC (ASCII 27) sale
            exit ( 0 );
            break;

        default:  //Si es cualquier otra tecla no hace nada
            break;
    }

    glutPostRedisplay();
}


void teclasFlechas ( int tecla, int x, int y )  // Funcion para manejo de teclas especiales (arrow keys)
{
  switch ( tecla ) {
    case GLUT_KEY_UP:  //gira sobre x sentido antihorario, valor positivo
        angleX +=2.0f;
        break;
    case GLUT_KEY_DOWN:  //gira sobre x sentido horario, valor negativo
        angleX -=2.0f;
        break;
    case GLUT_KEY_LEFT:  //gira sobre y sentido antihorario, valor positivo
        angleY +=2.0f;
        break;
    case GLUT_KEY_RIGHT:  //gira sobre y sentido horario, valor negativo
        angleY -=2.0f;
        break;
    default:
        break;
  }
  glutPostRedisplay();
}

void remodelar(int width, int height) {
    if (height == 0)
    {
        height = 1;
    }
    glViewport(0, 0, width, height);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    glFrustum(-5, 5, -5, 5, 3, 20.0);
    glutPostRedisplay();
}

void dibujar(void) {

    //glClearColor(0.035, 0.039, 0.043, 1);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
```

```
glLoadIdentity();
glTranslatef(transX,transY, transZ);
glRotatef(angleY,0.0,1.0,0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//Calzon
glScalef(6.0,2.0,1.0);
glRotatef(90.0,0.0,1.0,0.0);
prisma();

glLoadIdentity();
glTranslatef(transX, transY, transZ);
glRotatef(angleY, 0.0, 1.0, 0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//Pata derecha - primero escala luego traslada
glTranslatef(-1.75,-4.0f,0.0);
glScalef(2.5,6.0,1.0);
prisma();


glLoadIdentity();
glTranslatef(transX, transY, transZ);
glRotatef(angleY, 0.0, 1.0, 0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//Pata izquierda - primero escala luego traslada
glTranslatef(1.75, -4.0f, 0.0);
glScalef(2.5, 6.0, 1.0);
prisma();

glLoadIdentity();
glTranslatef(transX, transY, transZ);
glRotatef(angleY, 0.0, 1.0, 0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//Cinturon
glTranslatef(0.0, 1.25f, 0.0);
glScalef(6.0, .5, 1.0);
glRotatef(180.0, 0.0, 1.0, 0.0);
prisma();

glLoadIdentity();
glTranslatef(transX, transY, transZ);
glRotatef(angleY, 0.0, 1.0, 0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//bota derecha
glTranslatef(-1.75, -9.5f, 0.0);
glScalef(2.5, 5.0, 1.0);
glRotatef(90.0, 0.0, 1.0, 0.0);
prisma();

glLoadIdentity();
glTranslatef(transX, transY, transZ);
glRotatef(angleY, 0.0, 1.0, 0.0);
glRotatef(angleX, 1.0, 0.0, 0.0);

//bota izquierda
glTranslatef(1.75, -9.5f, 0.0);
glScalef(2.5, 5.0, 1.0);
glRotatef(90.0, 0.0, 1.0, 0.0);
prisma();
```