**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**“Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho”**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**

Imagen que contiene Diagrama

Descripción generada automáticamente

**TRABAJO DE TRANSFORMACIONES GEOMÉTRICAS**

**Docente:**

**JUAN CARLOS GUTIERREZ CACERES**

**Curso:**

**COMPUTACIÓN VISUAL**

**Alumno:**

**GRUPO 9**

**Velarde Huancahuari, Bryan Anthony**

**Arroyo Vasquez Luis Yazid**

**Salazar Garcia, Diego Alberto**

**Giovanni Arias Chumpitaz**

**LIMA – PERÚ**

**2024**

**Graficar un cubo que realice las siguientes transformaciones**FORWARD (f) Avanza hacia adelante 5 Unidades de distancia  
RIGHT (r) Gira a la derecha 10 Grados   
LEFT (l) Gira a la izquierda 10 Grados  
BACK (b) Retrocede 5 Unidades de distancia  
UPPITCH (u) Gira hacia arriba 10 Grados   
DOWNPITCH (d) Gira hacia abajo 10 Grados

|  |
| --- |
| #include <GL/glut.h>  GLfloat anguloCuboX = 0.0f;  GLfloat anguloCuboY = 0.0f;  GLfloat desp = 0.0f;  GLint ancho, alto;  int hazPerspectiva = 0;  void reshape(int width, int height)  {  glViewport(0, 0, width, height);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity(); //asegura de que cualquier transformación previa se elimine.  if(hazPerspectiva)  gluPerspective(60.0f, (GLfloat)width/(GLfloat)height, 1.0f, 20.0f);  else  glOrtho(-4, 4, -4, 4, 1, 10);    glMatrixMode(GL\_MODELVIEW);    ancho = width;  alto = height;  }  void drawCube(void)  {  glColor3f(1.0f, 0.0f, 0.0f);  glBegin(GL\_QUADS); //cara frontal  glVertex3f(-1.0f, -1.0f, 1.0f);  glVertex3f( 1.0f, -1.0f, 1.0f);  glVertex3f( 1.0f, 1.0f, 1.0f);  glVertex3f(-1.0f, 1.0f, 1.0f);  glEnd();    glColor3f(0.0f, 1.0f, 0.0f);  glBegin(GL\_QUADS); //cara trasera  glVertex3f( 1.0f, -1.0f, -1.0f);  glVertex3f(-1.0f, -1.0f, -1.0f);  glVertex3f(-1.0f, 1.0f, -1.0f);  glVertex3f( 1.0f, 1.0f, -1.0f);  glEnd();    glColor3f(0.0f, 0.0f, 1.0f);  glBegin(GL\_QUADS); //cara lateral izq  glVertex3f(-1.0f, -1.0f, -1.0f);  glVertex3f(-1.0f, -1.0f, 1.0f);  glVertex3f(-1.0f, 1.0f, 1.0f);  glVertex3f(-1.0f, 1.0f, -1.0f);  glEnd();  glColor3f(1.0f, 1.0f, 0.0f);  glBegin(GL\_QUADS); //cara lateral dcha  glVertex3f( 1.0f, -1.0f, 1.0f);  glVertex3f( 1.0f, -1.0f, -1.0f);  glVertex3f( 1.0f, 1.0f, -1.0f);  glVertex3f( 1.0f, 1.0f, 1.0f);  glEnd();    glColor3f(0.0f, 1.0f, 1.0f);  glBegin(GL\_QUADS); //cara arriba  glVertex3f(-1.0f, 1.0f, 1.0f);  glVertex3f( 1.0f, 1.0f, 1.0f);  glVertex3f( 1.0f, 1.0f, -1.0f);  glVertex3f(-1.0f, 1.0f, -1.0f);  glEnd();    glColor3f(1.0f, 0.0f, 1.0f);  glBegin(GL\_QUADS); //cara abajo  glVertex3f( 1.0f, -1.0f, -1.0f);  glVertex3f( 1.0f, -1.0f, 1.0f);  glVertex3f(-1.0f, -1.0f, 1.0f);  glVertex3f(-1.0f, -1.0f, -1.0f);  glEnd();    }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);    glLoadIdentity();    glTranslatef(0.0f, 0.0f, -5.0f);  glRotatef(anguloCuboX, 1.0f, 0.0f, 0.0f);  glRotatef(anguloCuboY, 0.0f, 1.0f, 0.0f);  glTranslatef(desp, 0.0f, 0.0f);    drawCube();    glFlush();  glutSwapBuffers();    }  void init()  {  glClearColor(0,0,0,0);  glEnable(GL\_DEPTH\_TEST);  ancho = 400;  alto = 400;  }  void idle()  {  display();  }  **void keyboard(unsigned char key, int x, int y)**  **{**  **switch(key)**  **{**  **case 'l':**  **anguloCuboY-=1.0f;**  **reshape(ancho,alto);**  **break;**    **case 'r':**  **anguloCuboY+=1.0f;**  **reshape(ancho,alto);**  **break;**    **case 'u':**  **anguloCuboX-=1.0f;**  **reshape(ancho,alto);**  **break;**    **case 'd':**  **anguloCuboX+=1.0f;**  **reshape(ancho,alto);**  **break;**    **case 'b':**  **reshape(ancho,alto);**  **desp -=0.5f;**  **break;**    **case 'f:**  **reshape(ancho,alto);**  **desp +=0.5;**  **break;**    **case 27: // escape**  **exit(0);**  **break;**    **}**  **}**  int main(int argc, char \*\*argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowPosition(100, 100);  glutInitWindowSize(ancho, alto);  glutCreateWindow("Cubo 1");  init();  glutDisplayFunc(display);  glutReshapeFunc(reshape);  glutIdleFunc(idle);  glutKeyboardFunc(keyboard);  glutMainLoop();  return 0;  } |

**Imágenes referenciales de los movimientos respecto a la perspectiva inicial:**

**PERESPECTIVA INICIAL**

**Imagen que contiene Icono

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “l”**

**Imagen que contiene Logotipo

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “r”**

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “u”**

**Imagen que contiene Forma

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “d”**

**Imagen que contiene Icono

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “b”**

**Imagen que contiene Forma

Descripción generada automáticamente**

**AL PRESIONAR LA TELCA “f”**

**Icono

Descripción generada automáticamente con confianza baja**