

UCS1712 – GRAPHICS AND MULTIMEDIA LAB

Lab Exercise 9: 3-Dimensional Projections in C++ using OpenGL

CODE:

```
#include<gl/glut.h>
#include<iostream>

using namespace std;

int alpha = 0, theta = 0;
bool flag = true;

void init() {
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glEnable(GL_DEPTH_TEST);
}

void keyPress(int key, int x, int y) {
    switch (key) {
        case GLUT_KEY_RIGHT: alpha++;
            break;
        case GLUT_KEY_LEFT:
            alpha--;
            break;
        case GLUT_KEY_UP:
            theta++;
            break;
        case GLUT_KEY_DOWN:
            theta--;
            break;
        case GLUT_KEY_HOME: flag = !flag;
            break;
    }
    glutPostRedisplay();
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (flag) glOrtho(-500, 500, -500, 500, -500, 500);
    else gluPerspective(100, 1, 0.1, 1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0, 0, 300, 0, 0, 0, 0, 0, 1, 0);
    glRotatef(alpha, 0, 1, 0);
    glRotatef(theta, 1, 0, 0);
    glColor3f(0.0, 0.0, 0.0);
    glutWireTorus(50, 150, 20, 20);
    glFlush();
}

int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
```

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);  
glutInitWindowSize(1000, 1000);  
glutCreateWindow("Parallel and Perspective Projections");  
init();  
glutDisplayFunc(display);  
glutSpecialFunc(keyPress);  
glutMainLoop();  
return 0;  
}
```

OUTPUT:



