UCS1712 – GRAPHICS AND MULTIMEDIA LAB

Ex. No.9 Parallel and Perspective projections

Date: 27/9/21 **Name:** Srinath S

Class: CSE-C **Roll:** 185001205

Question:

Write a menu driven program to perform Orthographic parallel projection and Perspective projection on any 3D object. Set the camera to any position on the 3D space.

Have (0,0,0) at the center of the screen.

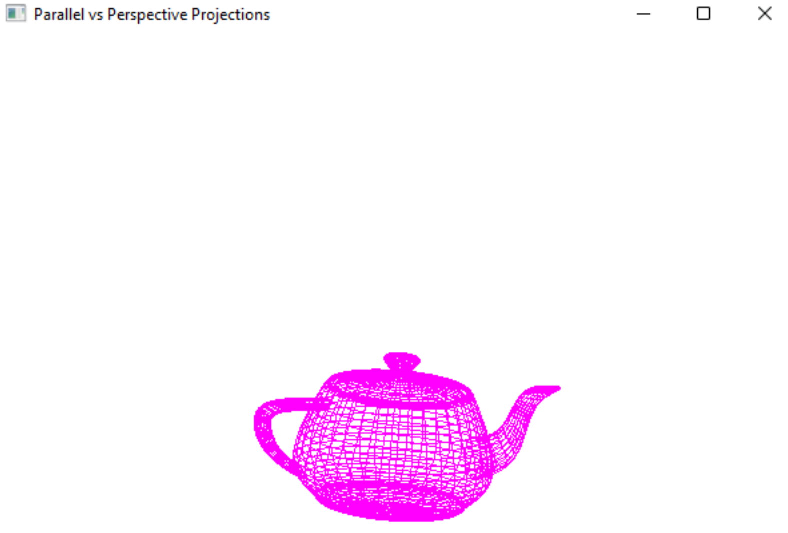
Draw X, Y and Z axis. You can use gluPerspective() to perform perspective projection.

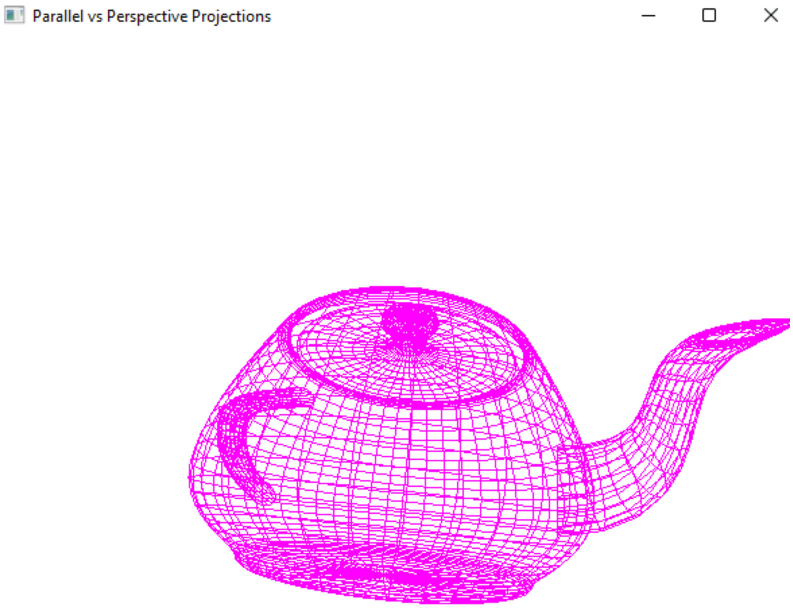
Use keyboard functions to rotate and show different views of the object. [Can use built-in functions for 3D transformations].

Code:

```
#include<iostream>
#include<math.h>
#include <GL/glut.h>

using namespace std;

int x = 0, y = 0;
int projType = 0;

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

void disp(int projType) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (projType == 1)
        gluPerspective(100, 1, 0.1, 100);
    else
        glOrtho(-2, 2, -2, 2, -2, 2);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0, 0, 1, 0, 0, 0, 0, 1, 0);
}

void display() {
    disp(projType);
    glRotatef(x, 0, 1, 0);
    glRotatef(y, 1, 0, 0);
    glColor3f(59.0, 0.0, 1.0);
    glutWireTeapot(0.5);
    glPopMatrix();
    glFlush();
}

void percieveKeyInterrupt(int key, int a, int b) {
    switch (key) {
    case GLUT_KEY_RIGHT: {
        x++;
        break;
    }

    case GLUT_KEY_LEFT: {
        x--;
        break;
    }

    case GLUT_KEY_UP: {
        y++;
        break;
    }

    case GLUT_KEY_DOWN: {
        y--;
        break;
    }

    }
    glutPostRedisplay();
}

void changeProjection(unsigned char c, int a, int b) {
    if (c == ' ') {
        projType ^= 1;
    }

    glutPostRedisplay();
}

int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(600, 600);
    glutCreateWindow("Parallel vs Perspective Projections");
    init();
    glutDisplayFunc(display);
    glutSpecialFunc(percieveKeyInterrupt);
    glutKeyboardFunc(changeProjection);
    glutMainLoop();
    return 0;
}
```

Outputs:


Parallel vs Perspective Projections


Parallel vs Perspective Projections