EX-09 - 3-Dimensional Projections in C++ using OpenGL

26/10/2021

Venkataraman Nagarajan, CSE - C
18500192

AIM

To implement 3d-projections in C++.

SPECIFICATION

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].

PROGRAM - 01

3D Projections

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

using namespace std;

int Xangle=0, Yangle=0;
int PROJECTION_FLAG = 0;

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

void disp(int i) {
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if(i == 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(PROJECTION_FLAG);

    glRotatef(Xangle, 0, 1, 0);
    glRotatef(Yangle, 1, 0, 0);

    glColor3f(1.0,0.0,0.0);
    glutWireTeapot(0.5);

    glPopMatrix();
    glFlush();
}

void percieveKeyInterrupt(int key, int x, int y) {
    switch(key) {
        case GLUT_KEY_RIGHT: {
            //glRotatef(2,1,0,0);
```

```
            Xangle++;
            break;
        }

        case GLUT_KEY_LEFT: {
            Xangle--;
            break;
        }

        case GLUT_KEY_UP: {
            Yangle++;
            break;
        }

        case GLUT_KEY_DOWN: {
            Yangle--;
            break;
        }

    }

        glutPostRedisplay();

}

void changeProjection(unsigned char c, int x, int y) {
    if(c == ' ') {
        PROJECTION_FLAG^=1;
    }

    glutPostRedisplay();
}

int main (int argc, char *argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
    glutInitWindowSize(600,600);
    glutCreateWindow("Projections");

    init();
    glutDisplayFunc(display);
    glutSpecialFunc(percieveKeyInterrupt);
    glutKeyboardFunc(changeProjection);
    glutMainLoop();
    return 0;
}
```

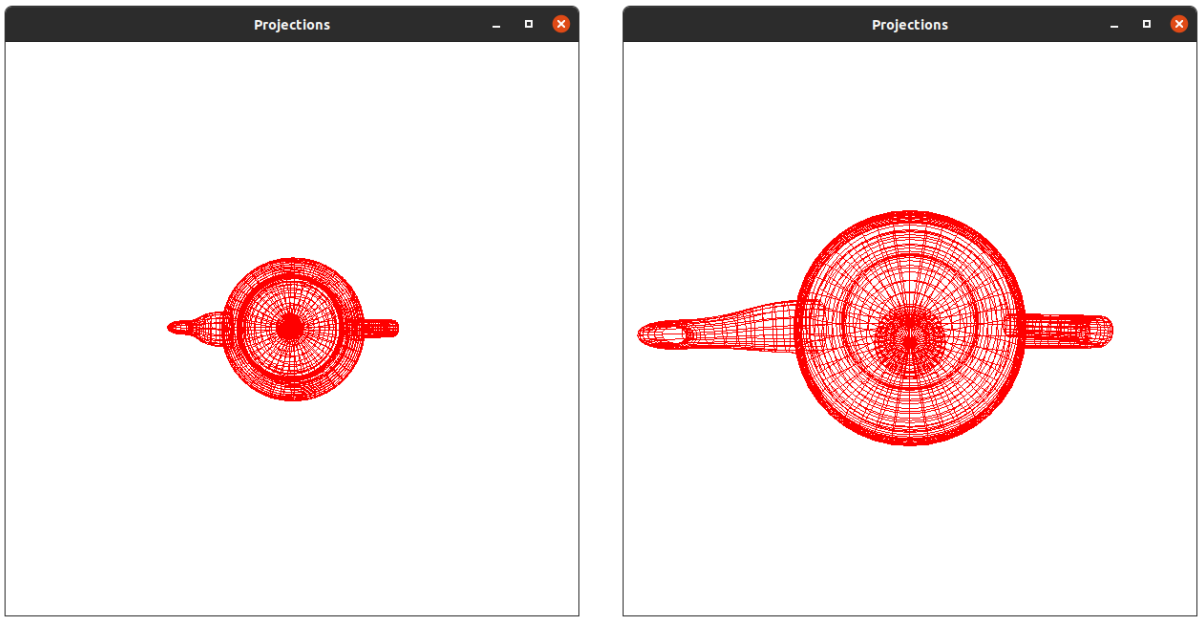

Figure 1: To the left is the parallel projection where Top and bottom part collapses into one, To the right is the perspective projection, where the size variation is clearly seen

SAMPLE I/0

RESULT

The code to implement 3d projections are written and output is verified.
