

GML Ex 9
Sabarivasan V
CSE-B
205001085

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 <GLUT/glut.h>
#include <iostream>
float angleX = 0.0, angleY = 0.0, angleZ = 0.0;
float cameraPosition[] = {5.0, 5.0, 10.0};
void drawAxes() {
    glColor3f(1.0, 0.0, 0.0); // Red X-axis
    glBegin(GL_LINES);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(5.0, 0.0, 0.0);
    glEnd();
    glColor3f(0.0, 1.0, 0.0); // Green Y-axis
    glBegin(GL_LINES);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(0.0, 5.0, 0.0);
    glEnd();
    glColor3f(0.0, 0.0, 1.0); // Blue Z-axis
    glBegin(GL_LINES);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(0.0, 0.0, 5.0);
    glEnd();
}
```

```

}
void drawCube() {
    glutWireCube(2.0); // You can replace this with your 3D
object drawing code
}
void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(cameraPosition[0], cameraPosition[1],
cameraPosition[2],
            0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    glRotatef(angleX, 1.0, 0.0, 0.0);
    glRotatef(angleY, 0.0, 1.0, 0.0);
    glRotatef(angleZ, 0.0, 0.0, 1.0);
    drawAxes();
    drawCube();
    glutSwapBuffers();
}
void reshape(int width, int height) {
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, (float)width / (float)height, 1.0,
100.0);
    glMatrixMode(GL_MODELVIEW);
}
void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        case 'x':
            angleX += 5.0;
            break;
        case 'X':
            angleX -= 5.0;
            break;
        case 'y':

```

```

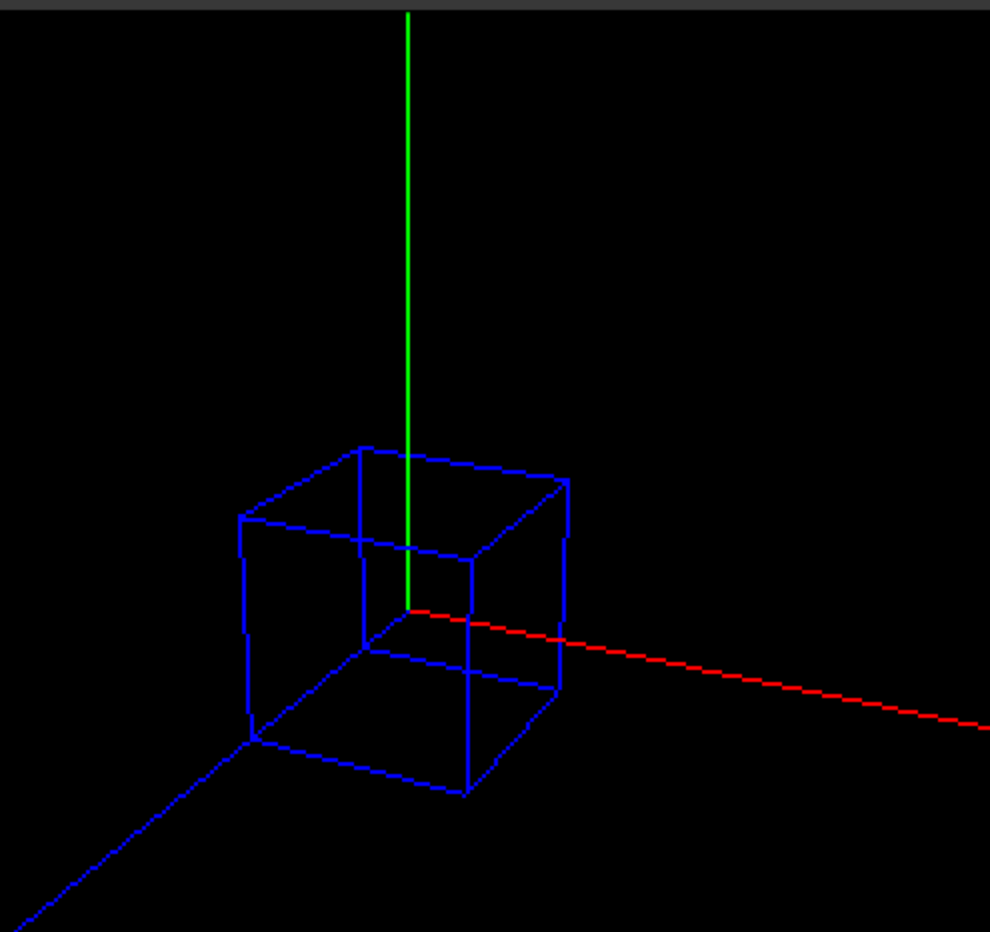
        angleY += 5.0;
        break;
    case 'Y':
        angleY -= 5.0;
        break;
    case 'z':
        angleZ += 5.0;
        break;
    case 'Z':
        angleZ -= 5.0;
        break;
    }
    glutPostRedisplay();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("3D Projection and Rotation");
    glEnable(GL_DEPTH_TEST);
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

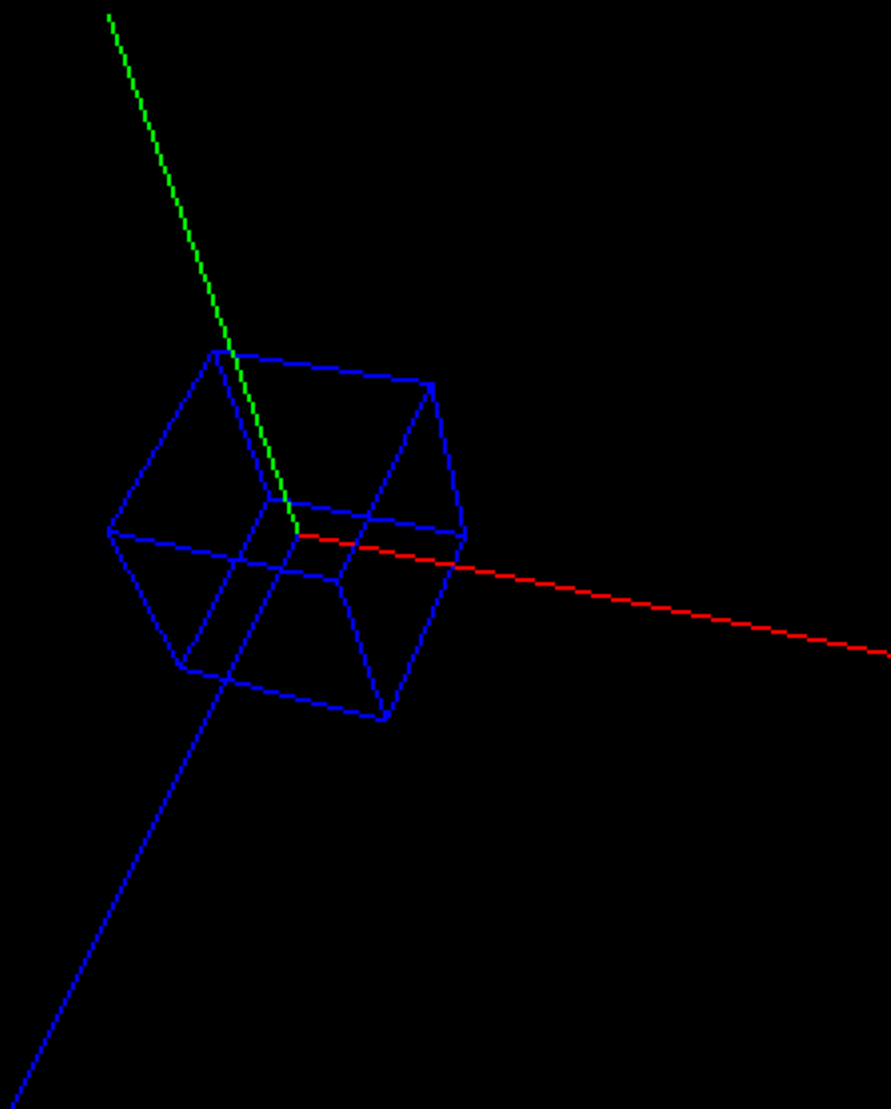
```

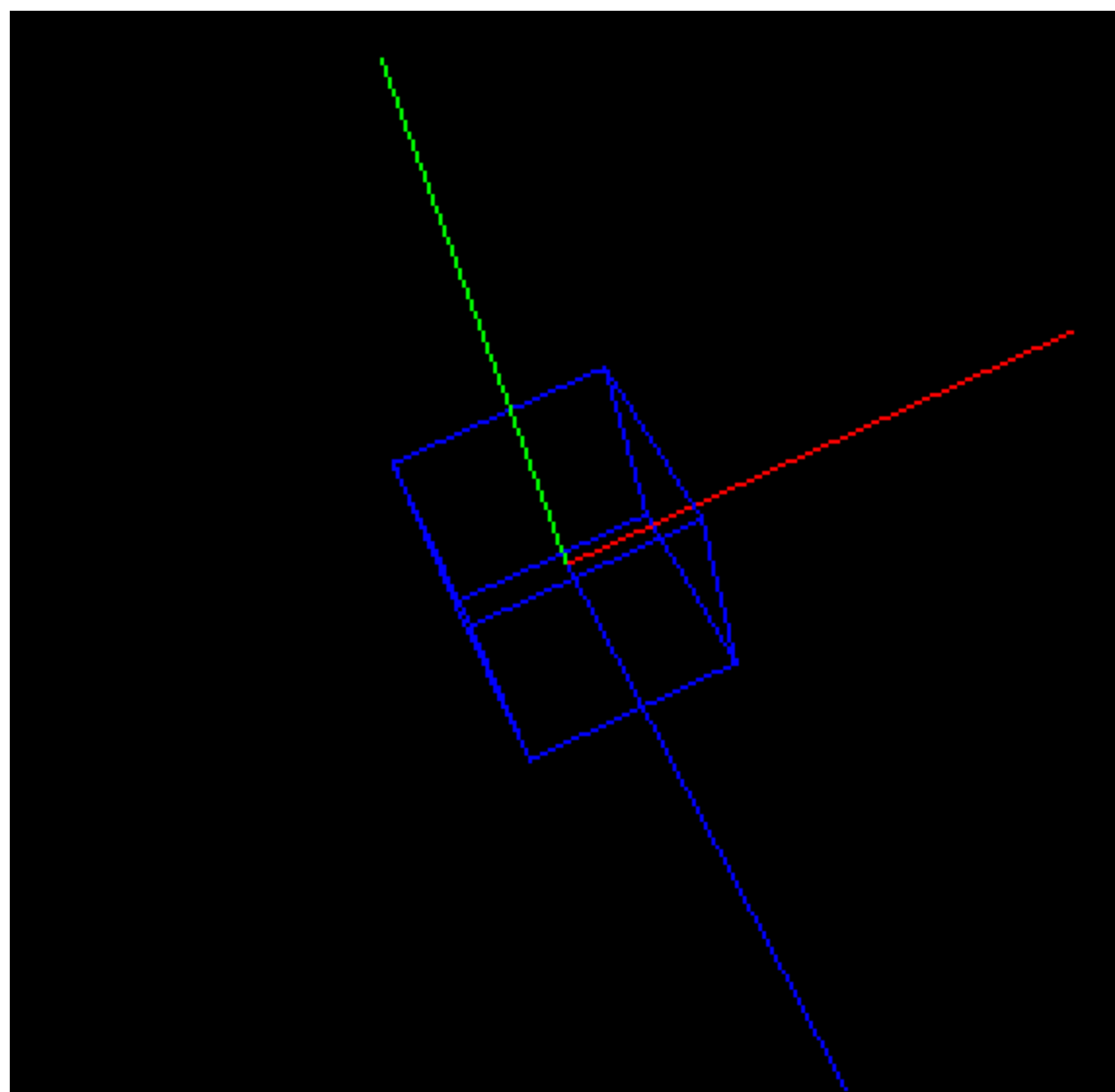
Output:

3D Projection and Rotation



3D Projection and Rotation







3D Projection and Rotation

