Name	**:**	**V Vikram**	**Class**	**:**	CSE 'C'
Reg. No.	**:**	**18 5001 194**	**Date**	**:**	27/10/2021
Subject	**:**	**UCS1712---Graphics and Multimedia Lab**			

QUESTION :

Lab Exercise 9

3-Dimensional Projections in C++ using OpenGL

- 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 axes. 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].

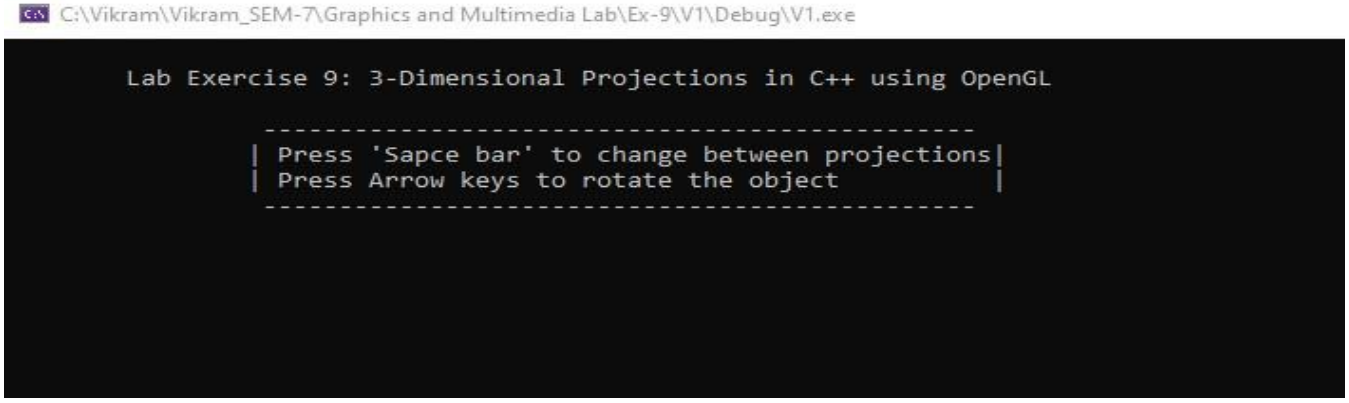
CODE :

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

using namespace std;

int X_ang = 0, Y_ang = 0;

int change_projection = 0;  //flag value for changing projection

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);   //Perspective projection
    else
        glOrtho(-2, 2, -2, 2, -2, 2);   //Orthographic parallel
projection
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    gluLookAt(0, 0, 1, 0, 0, 0, 0, 1, 0);
    /*creates a viewing matrix derived from an eye point,
    a reference point indicating the center of the scene, and an UP
vector.*/
}

void display() {
    disp(change_projection);

    glRotatef(X_ang, 0, 1, 0);
    glRotatef(Y_ang, 1, 0, 0);

    glColor3f(0.0, 0.0, 0.0);
    //void glutWireTorus(GLdouble innerRadius,GLdouble outerRadius,GLint
nsides, GLint rings);
    glutWireTorus(0.4, 0.8, 10, 30);

    glPopMatrix();
    /*pops the current matrix stack, replacing the
    current matrix with the one below it on the stack.*/

    glFlush();
}

//Gets key interrupt and rotates the object
void get_Key_input(int key, int x, int y) {
    switch (key) {
        case GLUT_KEY_UP: {
            Y_ang++;
            break;
        }
        case GLUT_KEY_DOWN: {
            Y_ang--;
            break;
        }
        case GLUT_KEY_RIGHT: {
            X_ang++;
            break;
        }

        case GLUT_KEY_LEFT: {
            X_ang--;
```

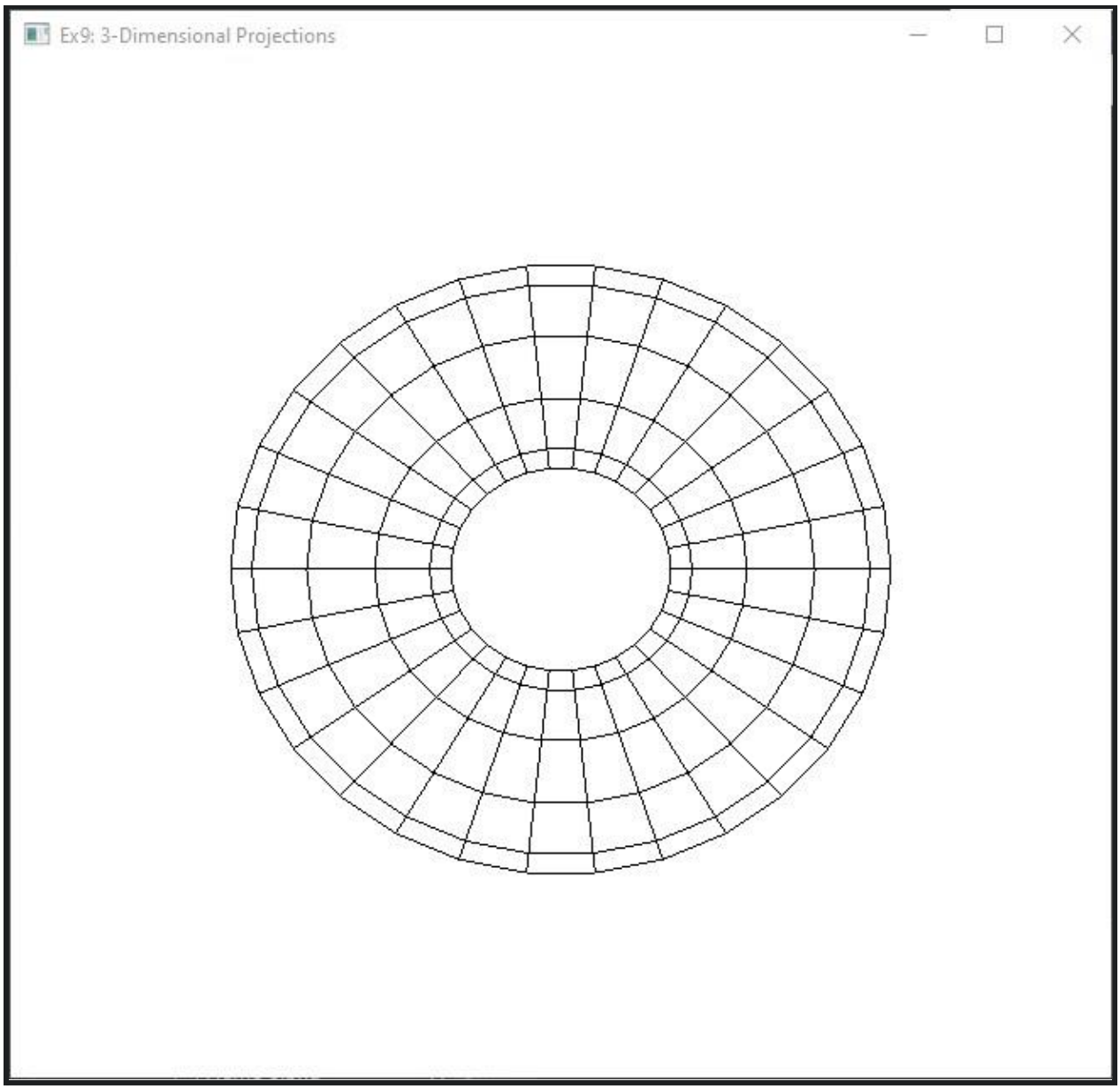
```
            break;
        }
    }
    glutPostRedisplay();
    //marks the current window as needing to be redisplayed

}

void changeProjection(unsigned char c, int x, int y) {
    /*
     On pressing "SPACE" change the projection between
     Orthographic parallel <--> Perspective .
    */
    if (c == ' ')   change_projection ^= 1;
    glutPostRedisplay();
}

int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(650, 600);
    glutCreateWindow("Ex9: 3-Dimensional Projections");
    cout << "\n\tLab Exercise 9: 3-Dimensional Projections in C++ using
OpenGL \n";
    cout << "\n\t\t -------------------------------------------------";
    cout << "\n\t\t| Press 'Sapce bar' to change between projections|";
    cout << "\n\t\t| Press Arrow keys to rotate the object          |";
    cout << "\n\t\t -------------------------------------------------";
    init();
    glutDisplayFunc(display);
    glutSpecialFunc(get_Key_input);
    glutKeyboardFunc(changeProjection);
    glutMainLoop();
    return 0;
}
```

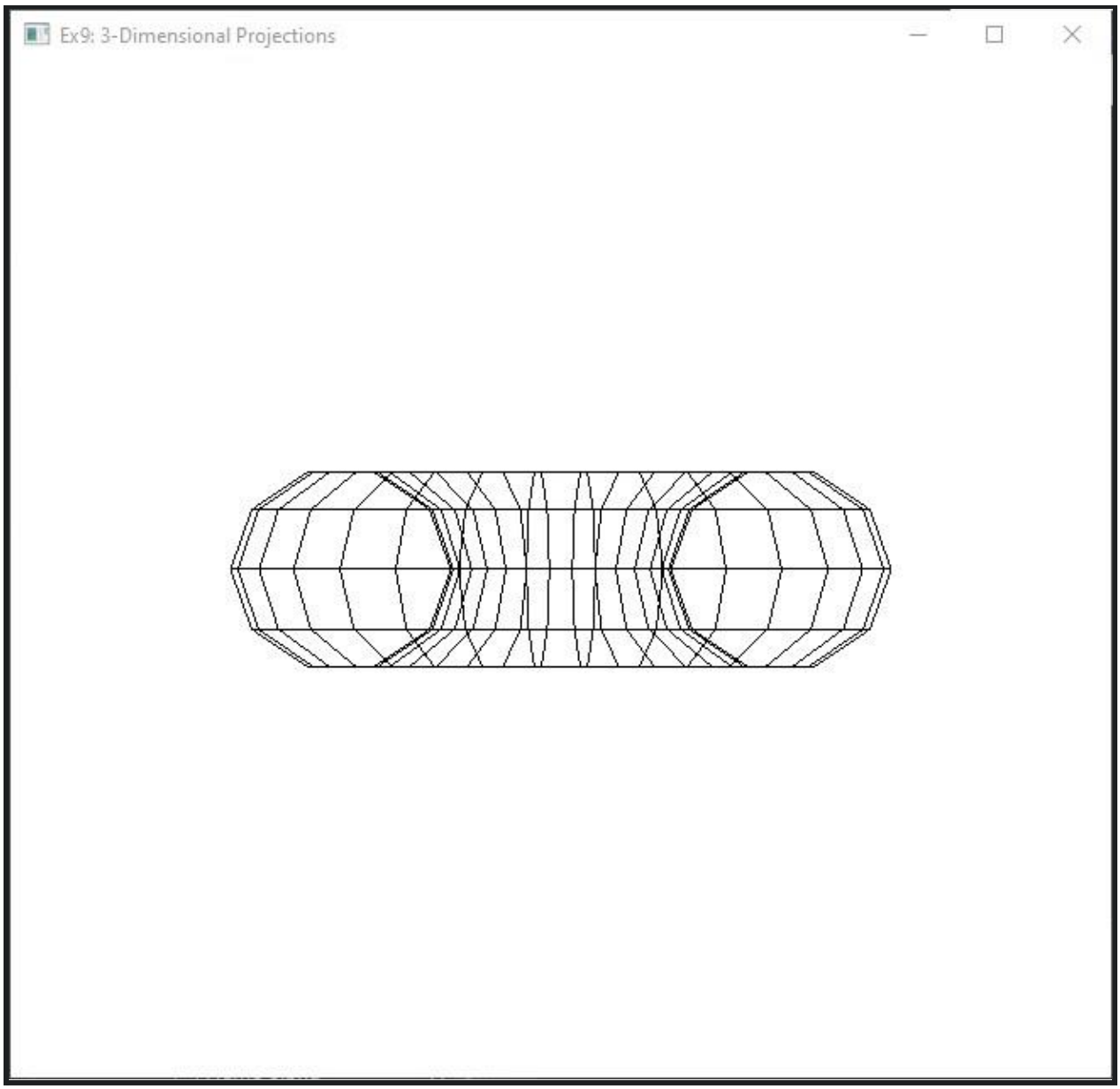
OUTPUT SNAPSHOTS :-

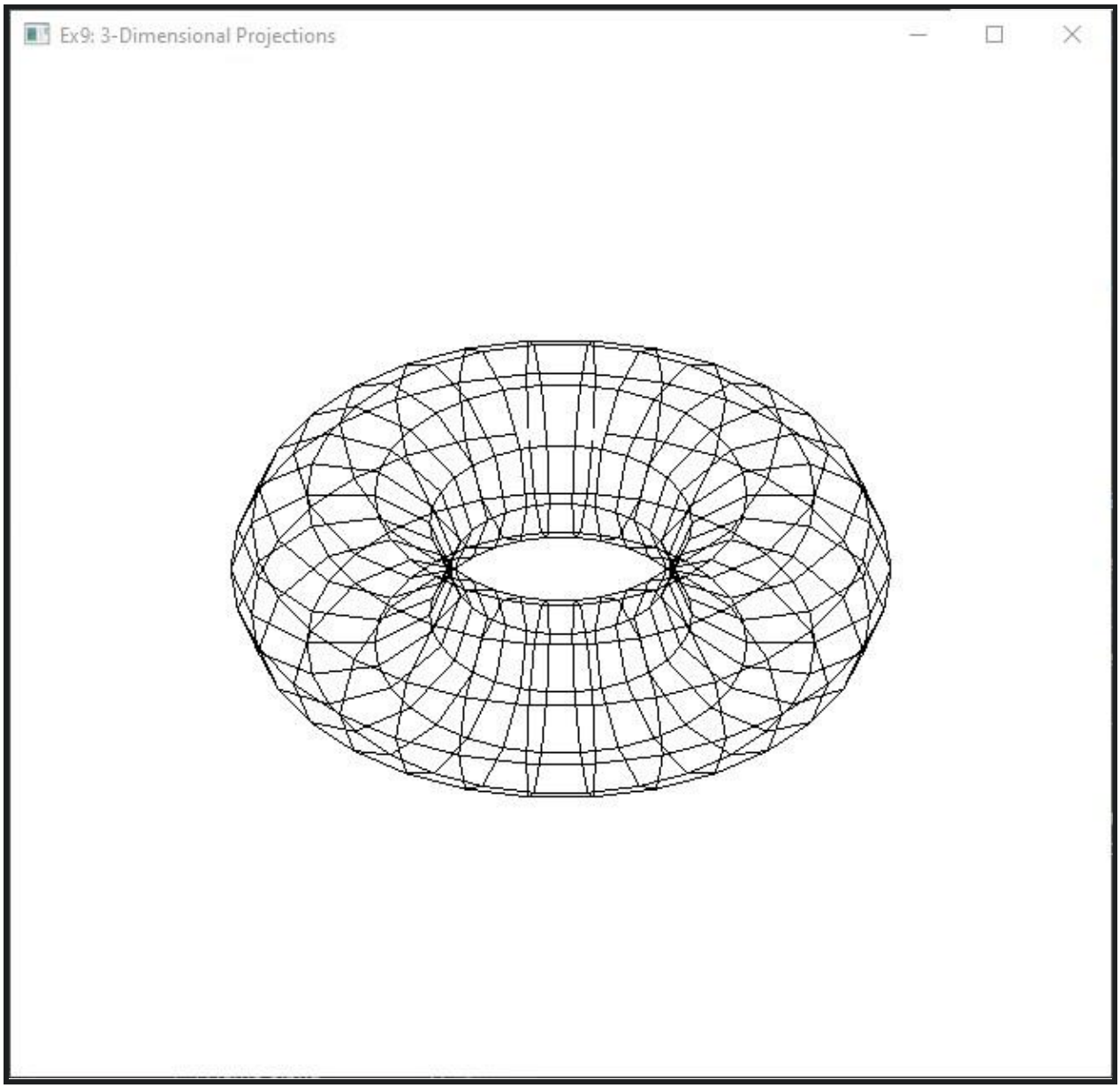


1.Orthographic Projection :-

Normal View :

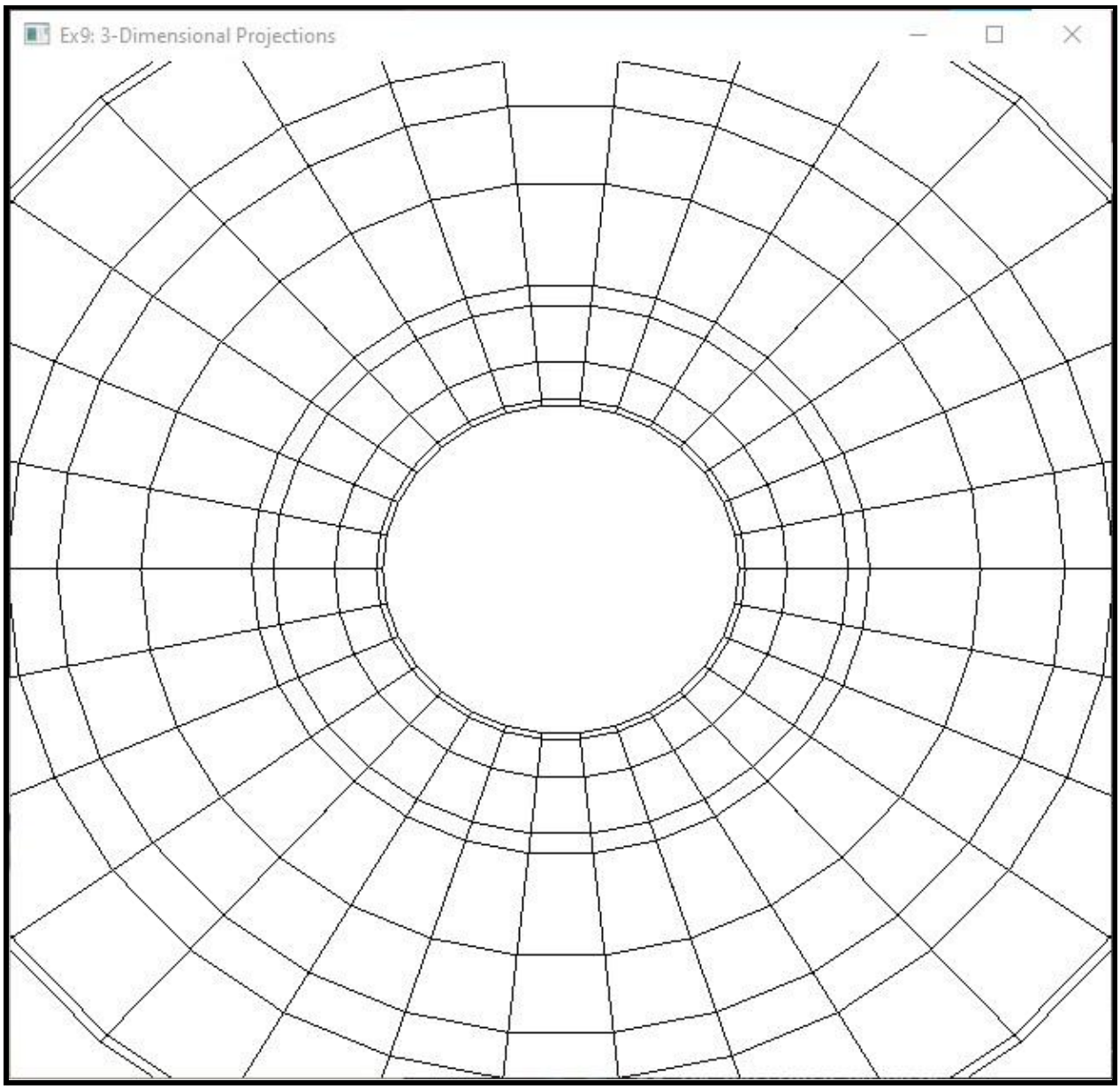


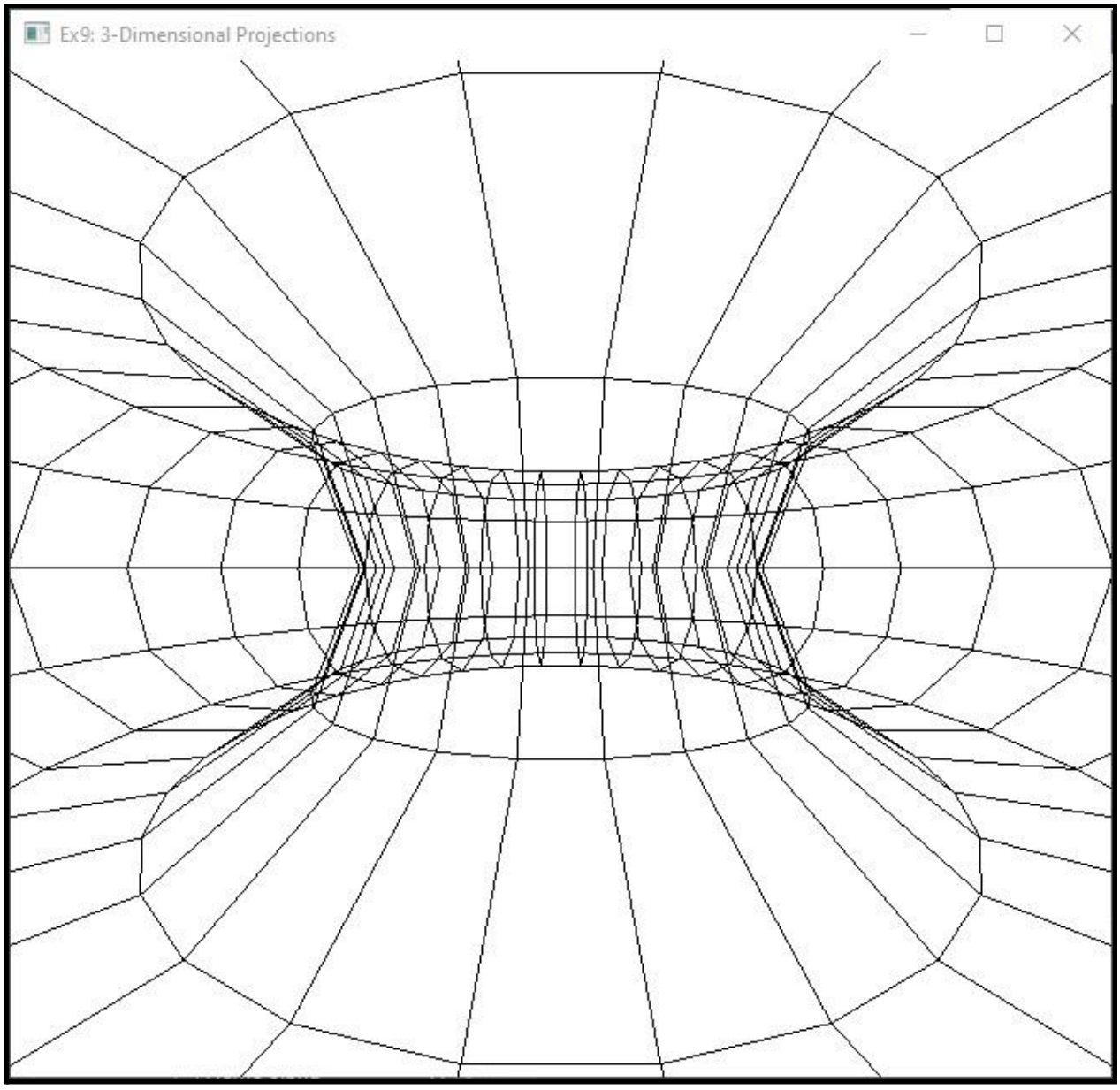

Ex9: 3-Dimensional Projections

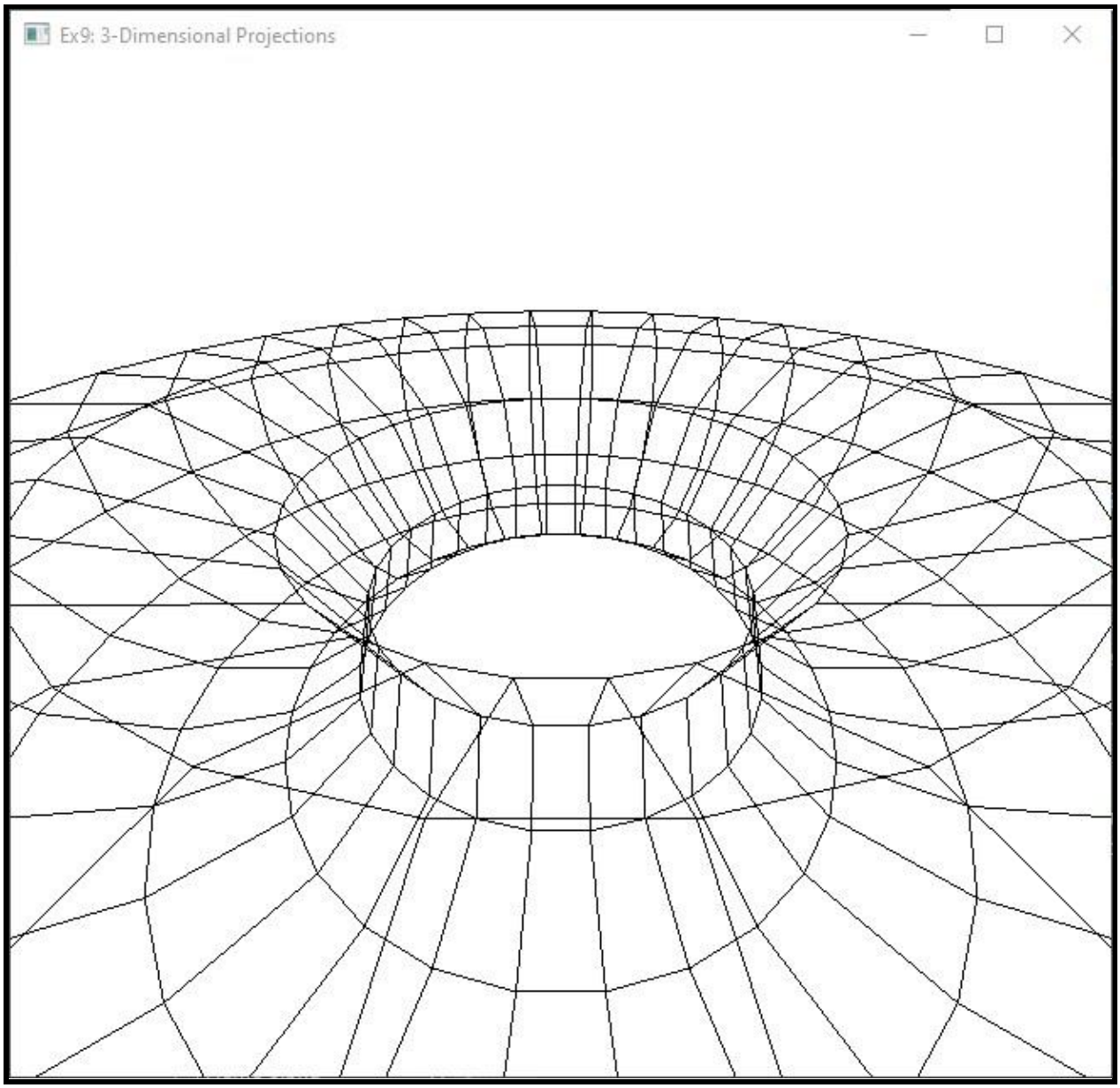

Ex9: 3-Dimensional Projections

2.Perspective Projection :-

Normal View :




Ex9: 3-Dimensional Projections


Ex9: 3-Dimensional Projections