SSN College of Engineering

Department of Computer Science and Engineering

## UCS1712 – GRAPHICS AND MULTIMEDIA LAB

EX NO: 9 – 3D Projections

Satheesh Kumar G R

185001136

AIM

To write a C++ program using OPENGL to perform 3D Projections – Orthographic and Perspective.

ALGORITHM:

1. Using glutWireTeapot() draw a teapot.
2. Draw axes for reference.
3. Space key is used to toggle between orthographic and perspective projections.
4. When in perspective projection mode:
   1. ‘w’ key pressed: X-rotate by +5
   2. ‘s’ key pressed: X-rotate by -5
   3. ‘a’ key pressed: Y-rotate by +5
   4. ‘d’ key pressed: Y-rotate by -5

CODE:

#include <iostream>

#include <cstring>

#include <GL/glut.h>

#include <math.h>

using namespace std;

const float WINDOW\_WIDTH = 1000;

const float WINDOW\_HEIGHT = 1000;

const float X\_MIN = -500;

const float X\_MAX = 500;

const float Y\_MIN = -500;

const float Y\_MAX = 500;

const int FPS = 60;

double x\_rotate = 0;

double y\_rotate = 0;

bool isOrthoProjection = true;

void initializeDisplay();

void keyboardKeys(unsigned char key, int x, int y);

void drawAxes();

int main(int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);

glutCreateWindow("3D Projections");

glutDisplayFunc(initializeDisplay);

glutKeyboardFunc(keyboardKeys);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glutMainLoop();

return 0;

}

void initializeDisplay()

{

//Initialize display parameters

glClearColor(1, 1, 1, 1);

glClear(GL\_COLOR\_BUFFER\_BIT);

//Translucency

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

//Line width

glLineWidth(3);

//Apply the transformations & drawing on the model view matrix

glMatrixMode(GL\_MODELVIEW);

//Draw the X and Y axis

drawAxes();

//Transform only the drawn object, so use the matrix stack accordingly

glPushMatrix();

if (isOrthoProjection)

{

//Parallel Projection

glOrtho(-2, 2, -2, 2, -2, 2);

}

else

{

//Perspective Projection

gluPerspective(120, 1, 0.1, 50); //FoVy = 120, Aspect Ratio = 1

}

gluLookAt(0, 0, 1, 0, 0, 0, 0, 1, 0); //Camera, Center & Up Vector

glRotatef(x\_rotate, 1, 0, 0); //Keyboard based rotations

glRotatef(y\_rotate, 0, 1, 0);

glColor4f(0, 0, 1, 0.3); //Draw the object

glutWireTeapot(0.5);

glPopMatrix(); //Pop the matrix back into the model view stack

glFlush();

}

void drawAxes()

{

//To draw X and Y axis

glColor3d(1, 0, 0);

glBegin(GL\_LINES);

glVertex2f(-2, 0);

glVertex2f(2, 0);

glVertex2f(0, -2);

glVertex2f(0, 2);

glEnd();

glFlush();

}

void keyboardKeys(unsigned char key, int x, int y)

{

//Callback function for keyboard interactivity

key = tolower(key);

switch (key)

{

case 'w':

{

x\_rotate += 5;

break;

}

case 's':

{

x\_rotate -= 5;

break;

}

case 'd':

{

y\_rotate += 5;

break;

}

case 'a':

{

y\_rotate -= 5;

break;

}

case 32:{

//Spacebar for changing projections

isOrthoProjection = !isOrthoProjection;

break;

}

}

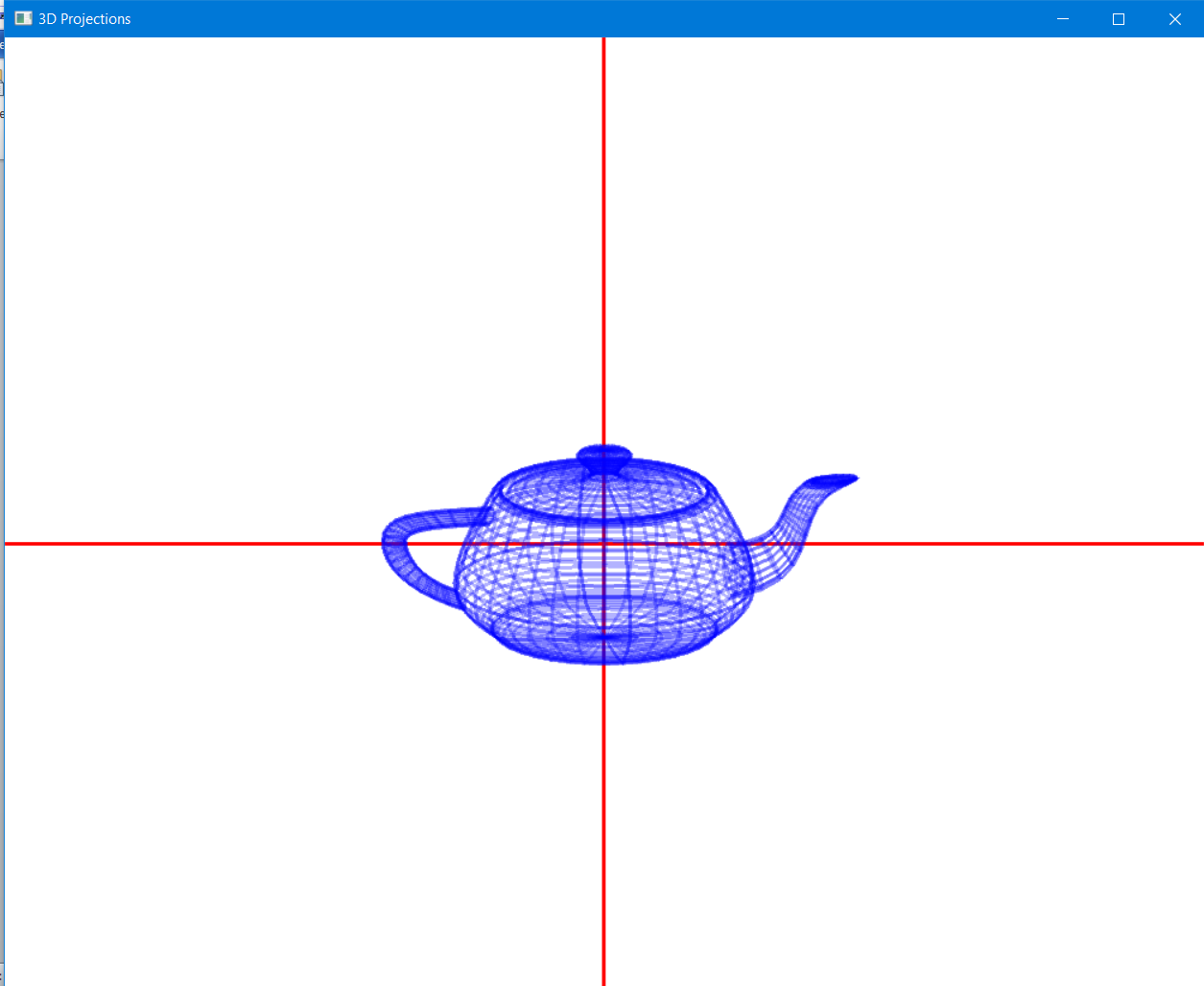
//Update the display

glutPostRedisplay();

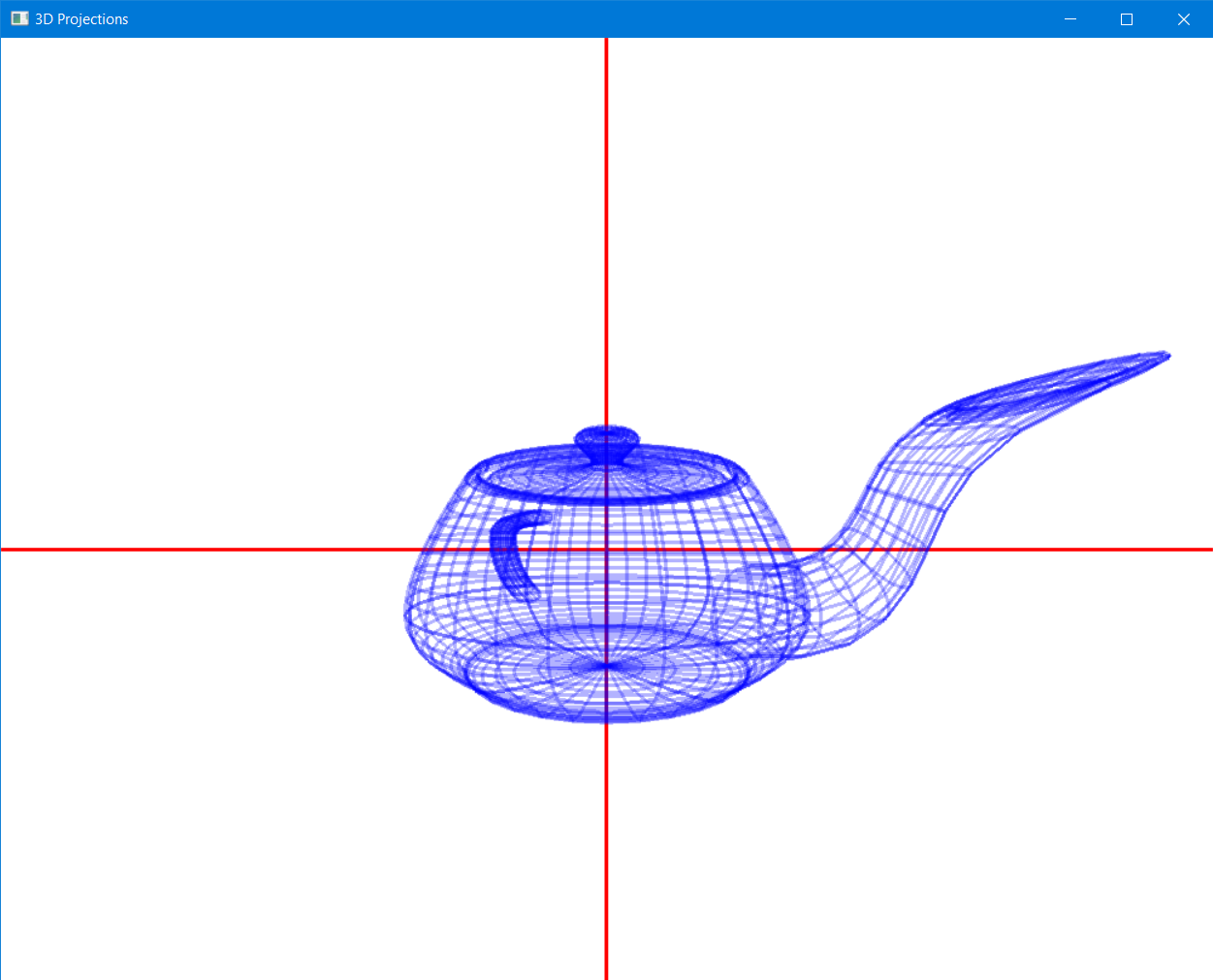
}

OUTPUT:

ORTHOGRAPHIC:



PERSPECTIVE:



RESULT:

OPENGL programs to perform 3D projections was designed and implemented successfully.