**UCS 1712 – GRAPHICS AND MULTIMEDIA LAB**

**ASSIGNMENT – 9**

**VISHAL N**

**185001198**

**25.10.2021 CSEC**

**1. PARALLEL AND PERSPECTIVE PROJECTIONS:**

**projections.cpp:**

#pragma warning(disable : 4996)

#include <GL/glut.h>

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

using namespace std;

bool\* keyStates = new bool[256];

int x\_angle = 0, y\_angle = 0, z\_angle = 0;

void drawAxis() {

glBegin(GL\_LINES);

glVertex3d(-500, 0, 0);

glVertex3d(500, 0, 0);

glEnd();

glBegin(GL\_LINES);

glVertex3d(0, 500, 0);

glVertex3d(0, -500, 0);

glEnd();

}

void drawTeapot() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

// X, Y and Z axis

glBegin(GL\_LINES);

glVertex3d(-5, 0, 0);

glVertex3d(5, 0, 0);

glVertex3d(0, -5, 0);

glVertex3d(0, 5, 0);

glVertex3d(0, 0, 1);

glVertex3d(0, 0, 100);

glEnd();

glColor3f(1.0, 0.0, 0.0);

glLoadIdentity();

glTranslatef(0.0f, 0.0f, -5.0f);

glPushMatrix();

glRotatef(x\_angle, 1, 0, 0);

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

glRotatef(z\_angle, 0, 0, 1);

glutWireCube(1);

glPopMatrix();

glFlush();

}

void keyOperations(void) {

int ANGLE\_INC = 45;

if (keyStates['w']) {

x\_angle += ANGLE\_INC;

}

else if (keyStates['s']) {

x\_angle -= ANGLE\_INC;

}

else if (keyStates['a']) {

y\_angle -= ANGLE\_INC;

}

else if (keyStates['d']) {

y\_angle += ANGLE\_INC;

}

else if (keyStates[' ']) {

z\_angle += ANGLE\_INC;

}

x\_angle %= 360;

y\_angle %= 360;

z\_angle %= 360;

drawTeapot();

}

void initialize() {

int WIDTH = 500, HEIGHT = 500, choice=1;

cout << "-----PROJECTIONS-----\n1 - Parallel Projection\n2 - Perspective Projection\nChoose any one projection: ";

cin >> choice;

glClearColor(1.0f, 1.0f, 1.0f, 0.0f);

glViewport(0, 0, WIDTH, HEIGHT);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if (choice == 1) {

glOrtho(-2.0, 2.0, -2.0, 2.0, 1, 100);

}

else {

gluPerspective(60, (GLfloat)WIDTH / (GLfloat)HEIGHT, 1, 100.0);

}

glMatrixMode(GL\_MODELVIEW);

for (int i = 0; i < 256; i++) {

keyStates[i] = false;

}

}

void keyPressed(unsigned char key, int x, int y) {

keyStates[key] = true;

keyOperations();

}

void keyUp(unsigned char key, int x, int y) {

keyStates[key] = false;

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutCreateWindow("Projections");

glutDisplayFunc(drawTeapot);

drawAxis();

initialize();

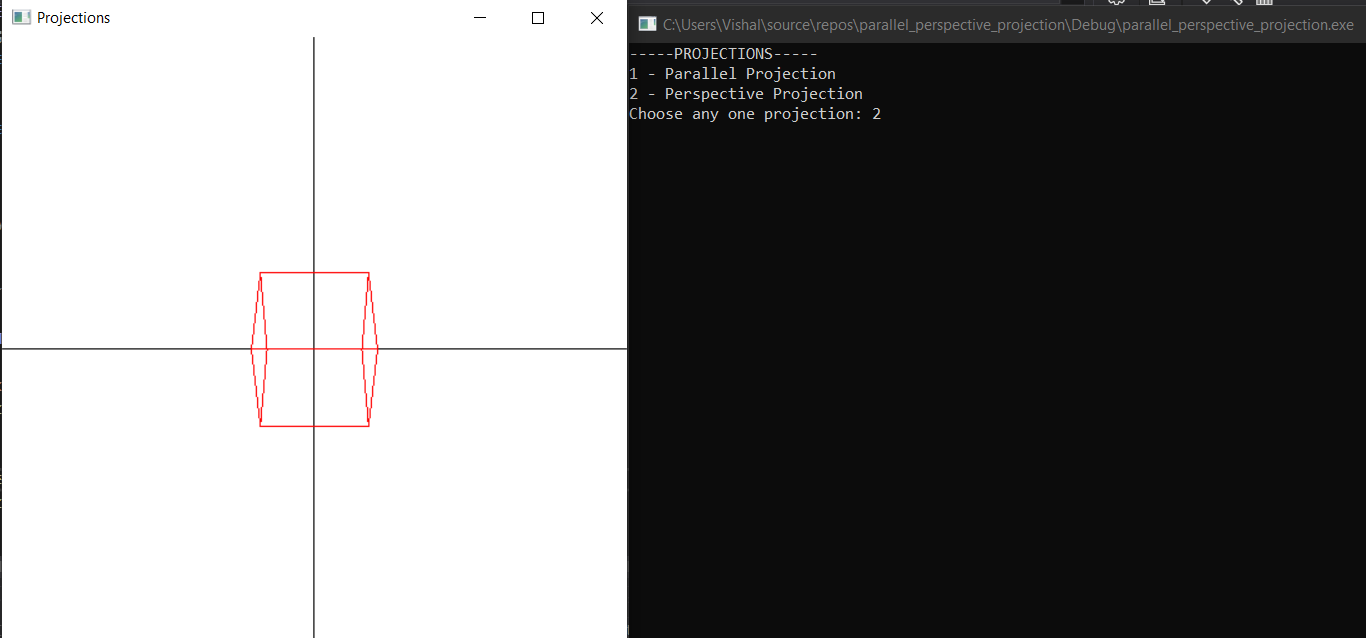
glutKeyboardFunc(keyPressed);

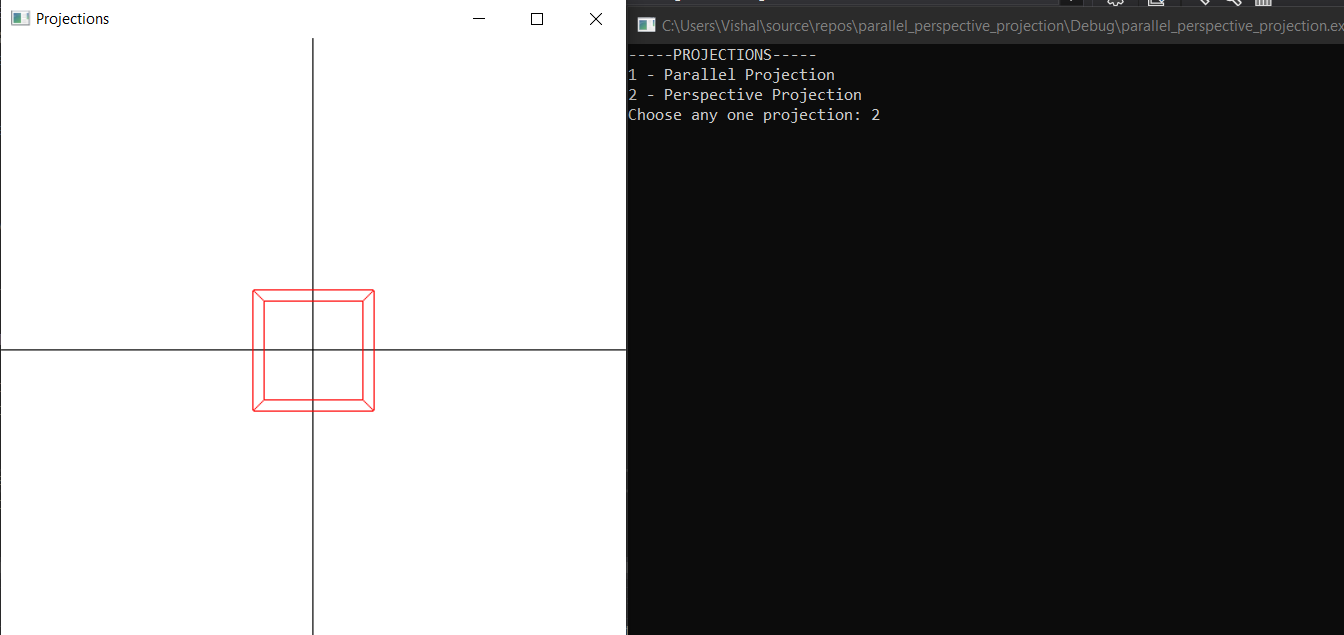
glutKeyboardUpFunc(keyUp);

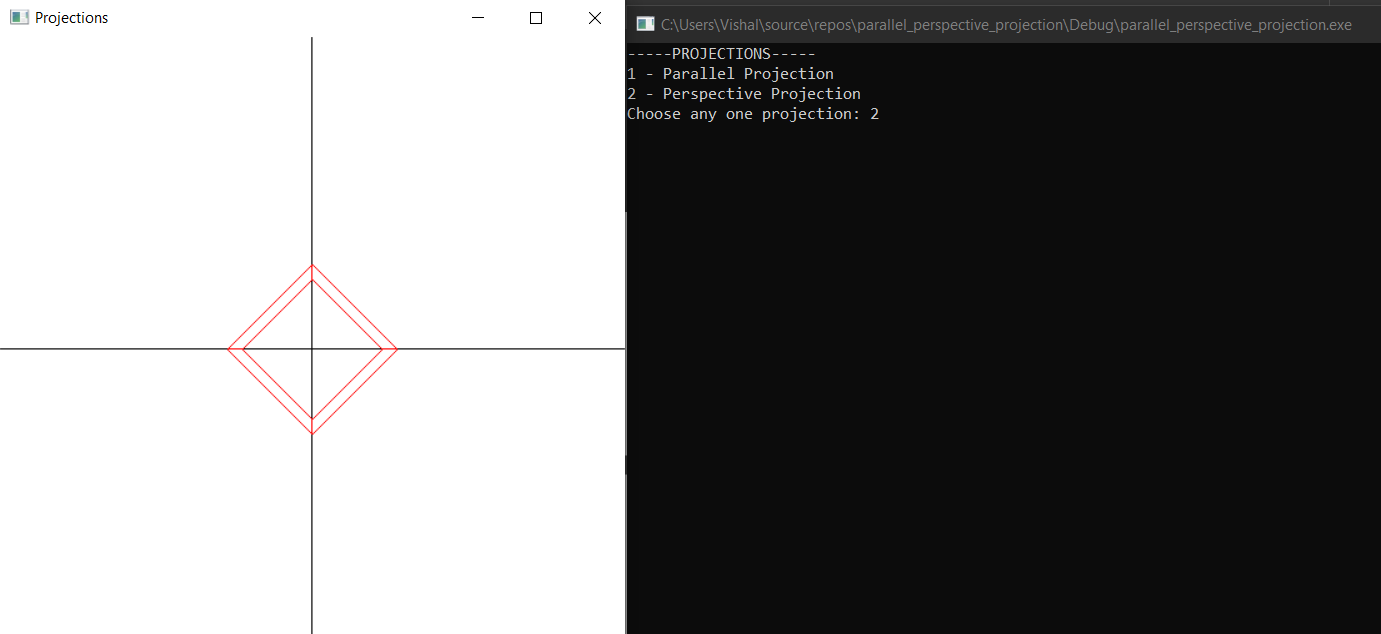
glutMainLoop();

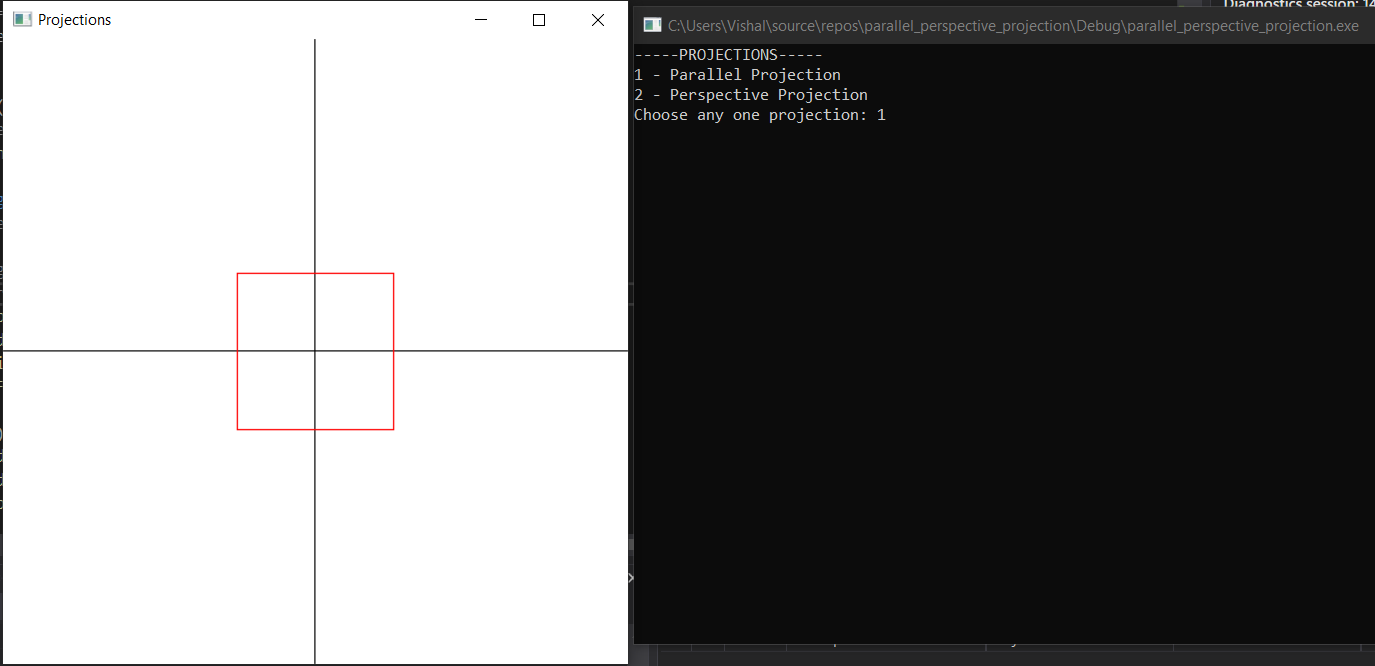
}

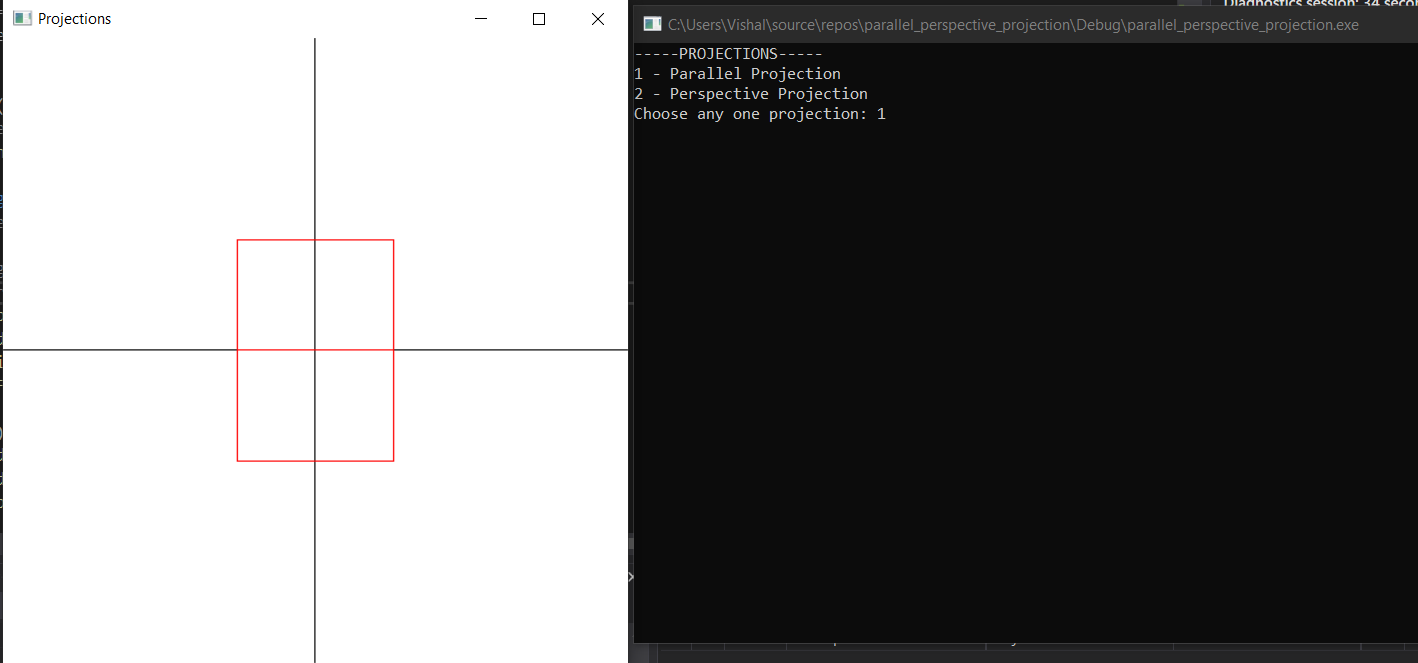
**OUTPUTS:**

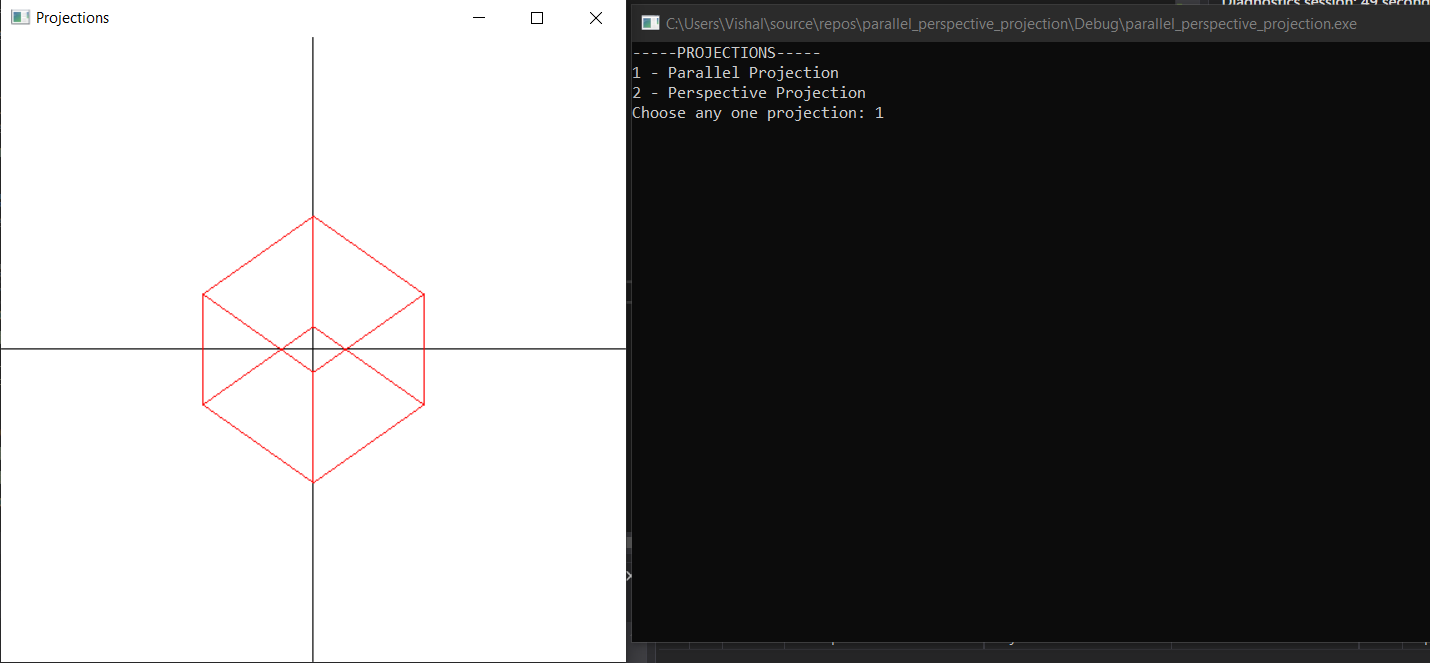
****

****

****

****

****

****