**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**Gokhulnath T**

**185001051**

------------------------------------------------------------------------------------------------------------

Lab Exercise 1 : Study of Basic Output Primitives in C++ using OpenGL

a). To create an output window using OPENGL and to draw the following basic output primitives –

* POINTS

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_POINTS);

    glVertex2d(150, 100);

    glVertex2d(100, 230);

    glVertex2d(170, 130);

    glVertex2d(300, 350);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

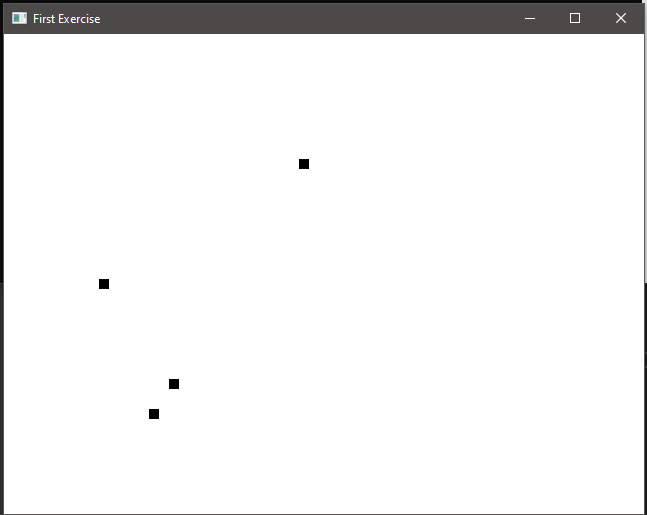
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* LINES

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_LINES);

    glVertex2d(180, 15);

    glVertex2d(10, 145);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

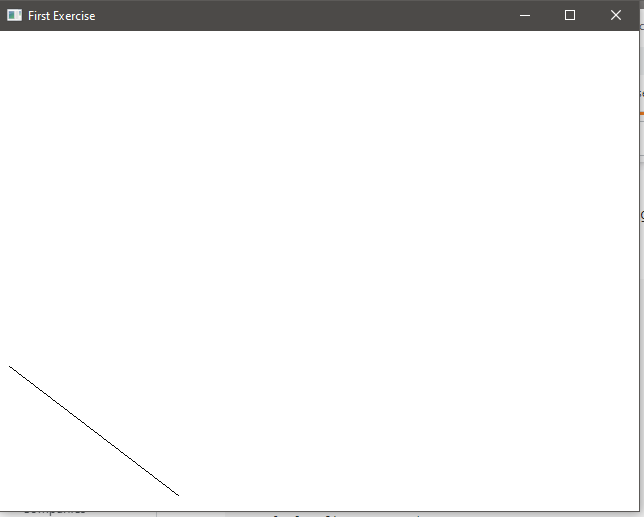
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* LINE\_STRIP

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_LINE\_STRIP);

    glVertex2f(0, 0);

    glVertex2f(50, 100);

    glVertex2f(50, 100);

    glVertex2f(100, 0);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

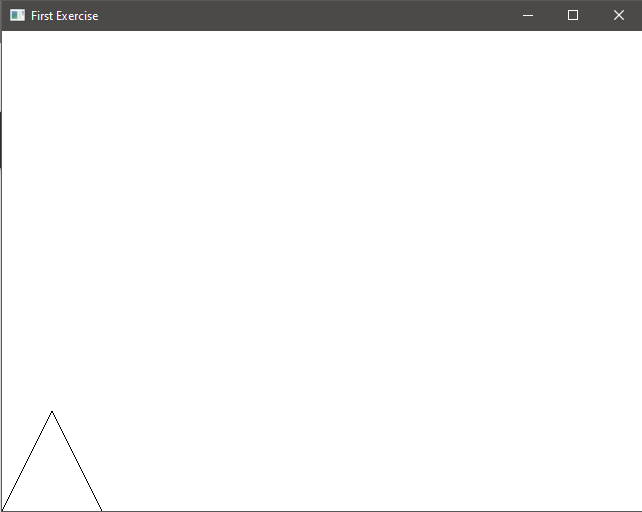
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* LINE\_LOOP

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_LINE\_LOOP);

    glVertex2f(50, 50);

    glVertex2f(100, 100);

    glVertex2f(100, 150);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

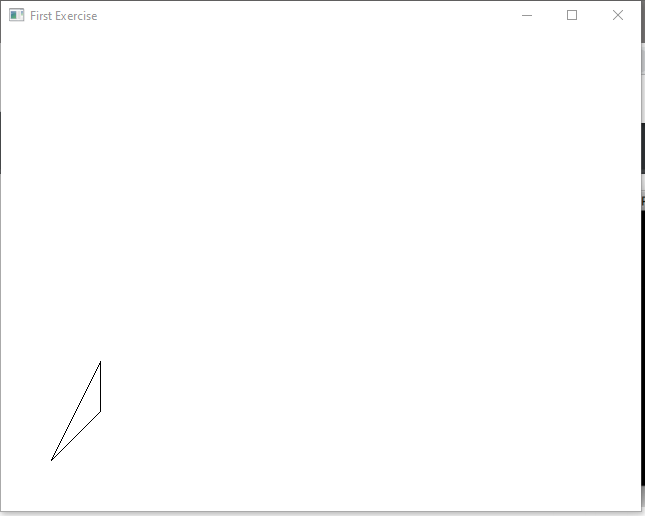
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* TRIANGLES

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_TRIANGLES);

    glVertex2f(50, 50);

    glVertex2f(75, 200);

    glVertex2f(150, 50);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

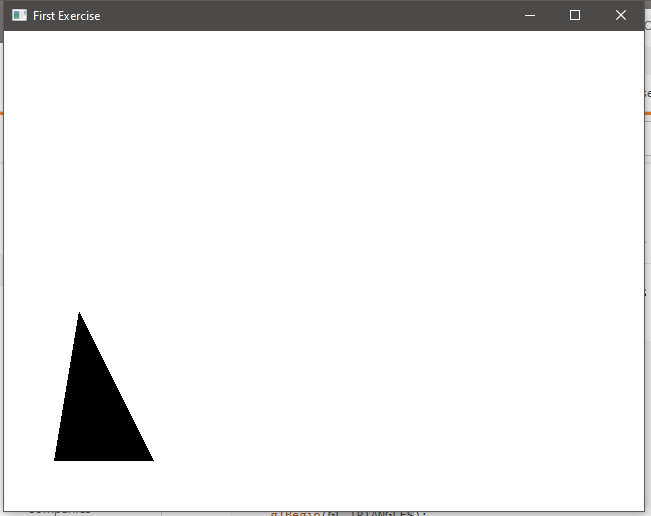
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* QUADS

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_QUADS);

    glVertex2f(50, 50);

    glVertex2f(75, 200);

    glVertex2f(150, 150);

    glVertex2f(200, 75);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

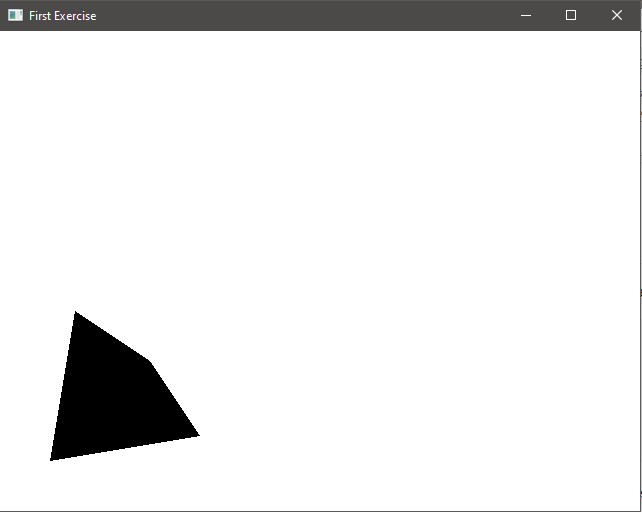
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



* QUAD\_STRIP

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    for (int j = 1; j <= 8; j++) {

        for (int i = 1; i <= 8; i ++) {

            glBegin(GL\_QUADS);

            if ((i + j) % 2 == 0) {

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f(i \* 50, j \* 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f(i \* 50, (j \* 50) + 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f((i \* 50) + 50, (j \* 50) + 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f((i \* 50) + 50, j \* 50);

            }

            else {

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f(i \* 50, j \* 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f(i \* 50, (j \* 50) + 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f((i \* 50) + 50, (j \* 50) + 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f((i \* 50) + 50, j \* 50);

            }

            glEnd();

        }

    }

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

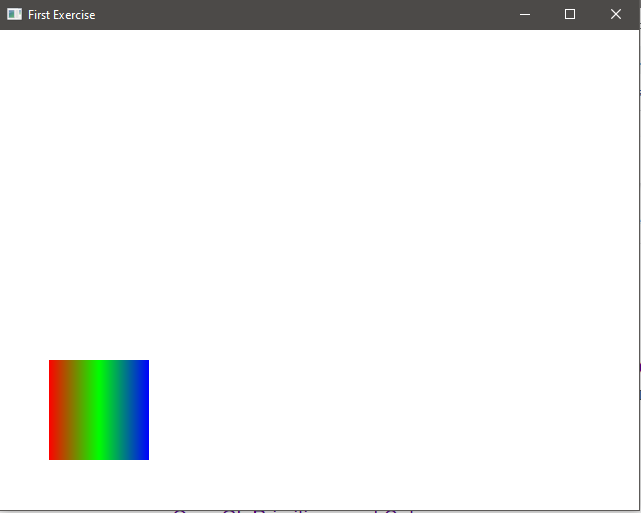
    glutCreateWindow("First Exercise");

    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}

* POLYGON

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_POLYGON);

    glVertex2f(50, 50);

    glVertex2f(50, 150);

    glVertex2f(100, 150);

    glVertex2f(200, 250);

    glVertex2f(150, 50);

    glVertex2f(150, 150);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

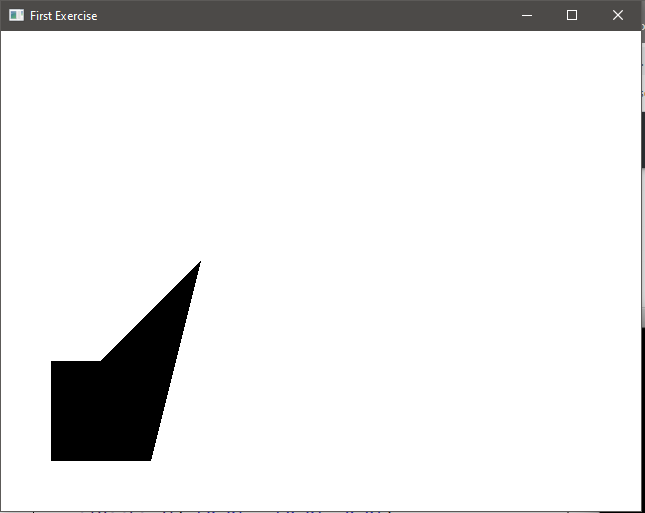
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



1. To create an output window and draw a checkerboard using OpenGL.

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    for (int j = 1; j <= 8; j++) {

        for (int i = 1; i <= 8; i ++) {

            glBegin(GL\_QUADS);

            if ((i + j) % 2 == 0) {

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f(i \* 50, j \* 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f(i \* 50, (j \* 50) + 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f((i \* 50) + 50, (j \* 50) + 50);

                glColor3f(0.0f, 0.0f, 0.0f);

                glVertex2f((i \* 50) + 50, j \* 50);

            }

            else {

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f(i \* 50, j \* 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f(i \* 50, (j \* 50) + 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f((i \* 50) + 50, (j \* 50) + 50);

                glColor3f(1.0f, 1.0f, 1.0f);

                glVertex2f((i \* 50) + 50, j \* 50);

            }

            glEnd();

        }

    }

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("First Exercise");

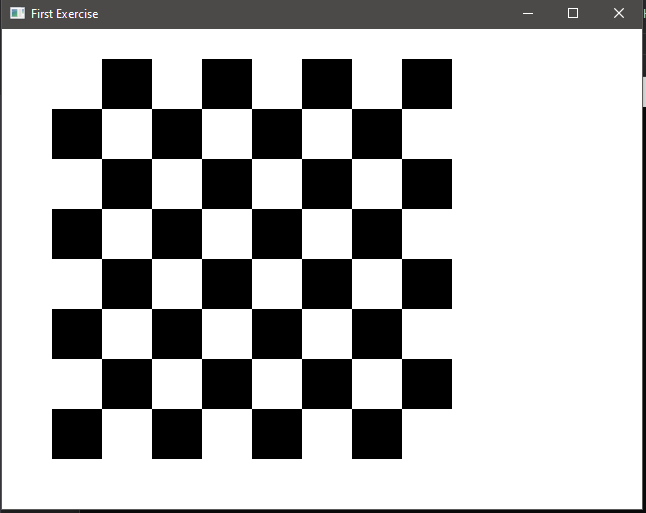
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}



1. To create an output window and draw a house using POINTS,LINES,TRAINGLES and QUADS/POLYGON.

#include<windows.h>

#include<stdio.h>

#include<gl/glut.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void myDisplay() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

*//Roof triangle*

    glBegin(GL\_TRIANGLES);

    glVertex2f(200, 200);

    glVertex2f(250, 300);

    glVertex2f(300, 200);

    glEnd();

*//Base block*

    glBegin(GL\_QUADS);

    glVertex2f(200, 100);

    glVertex2f(200, 200);

    glVertex2f(300, 200);

    glVertex2f(300, 100);

    glEnd();

*//door*

    glBegin(GL\_QUADS);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex2f(240, 100);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex2f(240, 130);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex2f(260, 130);

    glColor3f(1.0f, 1.0f, 1.0f);

    glVertex2f(260, 100);

    glEnd();

*//road*

    glBegin(GL\_LINES);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(0, 100);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(640, 100);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(0, 10);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(640, 10);

*//dividers*

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(50, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(150, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(200, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(300, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(350, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(450, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(500, 55);

    glColor3f(0.0f, 0.0f, 0.0f);

    glVertex2d(600, 55);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

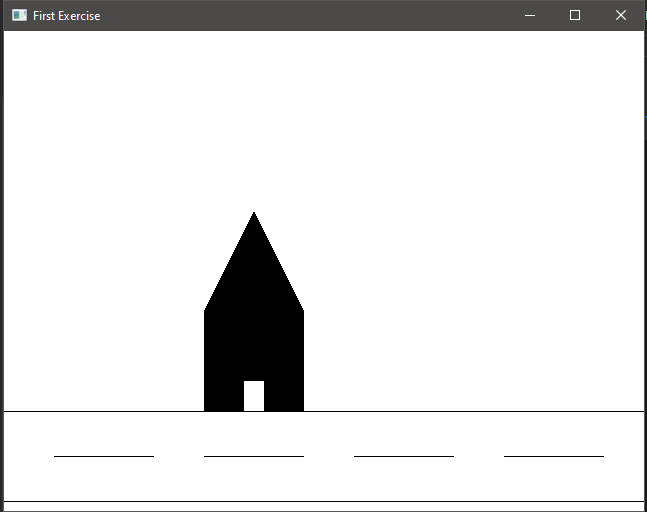
    glutCreateWindow("First Exercise");

    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}