EX-01 - Basic Output Primitives using C++ with OpenGL

16/07/2021

Venkataraman Nagarajan, CSE - C
18500192

AIM

To write and check out the output primitives in C++.

SPECIFICATION

- Create an output window using OpenGL and to draw the following basic output primitives – POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, QUADS, QUAD_STRIP, POLYGON.
- Create an output window and draw a checkerboard using OpenGL.
- Create an output window and draw a house using POINTS,LINES,TRIANGLES and QUADS/POLYGON.

PROGRAM - 01

Checking out output primitives

```c
// Q: To create an output window using OPENGL and to draw the following basic output primitives
//     - POINTS
//     - LINES
//     - LINE_STRIP
//     - LINE_LOOP
//     - TRIANGLES
//     - QUADS
//     - QUAD_STRIP
//     - POLYGON.

// Reference: https://docs.microsoft.com/en-us/windows/win32/opengl/glbegin

#include<GL/glut.h>

const int WINDOW_WIDTH = 850;
const int WINDOW_HEIGHT = 700;

void myInit();
void myDisplay();

void printPoints();
void printLines();
void printLineStrip();
void printLineLoop();
void printTriangles();
void printQuads();
void printQuadStrip();
void printPolygon();

int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
    glutCreateWindow("Basic Shapes");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}

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);

    printPoints();
    printLines();
    printLineStrip();
    printLineLoop();
    printTriangles();
    printQuads();
    printQuadStrip();
    printPolygon();

    glFlush();
}

void printPoints() {
    //1 - pt1, 2 - pt2, 3 - pt3, ...

    glBegin(GL_POINTS);

    glVertex2d(10,10);    //1
    glVertex2d(20,20);    //2
    glVertex2d(20,10);    //3
    glVertex2d(10,20);    //4

    glEnd();
}

void printLines() {
    //1-2 Line

    glBegin(GL_LINES);

    glVertex2d(30,30);    //1
    glVertex2d(40,40);    //2

    glVertex2d(30,40);
    glVertex2d(40,30);

    glEnd();
}

void printLineStrip() {
    //1-2-3-4-..-(n-1)-n Lines

    glBegin(GL_LINE_STRIP);

    glVertex2d(50,50);    //1
    glVertex2d(60,50);    //2
    glVertex2d(60,60);    //3
    glVertex2d(50,60);    //4

    glEnd();
}

void printLineLoop() {
    //1-2-3-4-....-n-1 Lines

    glBegin(GL_LINE_LOOP);

    glVertex2d(70,70);   //1
    glVertex2d(80,70);   //2
    glVertex2d(80,80);   //3
    glVertex2d(70,80);   //4

    glEnd();
}

void printTriangles() {
    //1-2-3-1 Triangle

    glBegin(GL_TRIANGLES);

    glVertex2d(90,90);    //1
    glVertex2d(90,100);   //2
    glVertex2d(100,90);   //3

    glVertex2d(100,110);
    glVertex2d(90,110);
    glVertex2d(100,100);

    glEnd();
}

void printQuads() {
    //1-2-3-4-1 Quad

    glBegin(GL_QUADS);

    glVertex2d(110,110);  //1
    glVertex2d(120,110);  //2
    glVertex2d(120,120);  //3
    glVertex2d(110,120);  //4

    glVertex2d(130,130);
    glVertex2d(140,130);
    glVertex2d(140,150);
    glVertex2d(130,150);

    glEnd();
}

void printQuadStrip() {
    //1-2-4-3 Quad1
    //3-4-6-5 Quad2

    glBegin(GL_QUAD_STRIP);

    glVertex2d(150,150);       //1
    glVertex2d(160,150);       //2

    glVertex2d(150,160);       //3
    glVertex2d(160,160);       //4

    glVertex2d(150,170);       //5
    glVertex2d(160,170);       //6

    glEnd();
}

void printPolygon() {
    //(1,2,3,4,..,n) Convex polygon

    glBegin(GL_POLYGON);

    glVertex2d(180,180);       //1
    glVertex2d(200,180);       //2
    glVertex2d(220,190);       //3
    glVertex2d(220,210);       //4
    glVertex2d(200,230);       //5
    glVertex2d(180,230);       //6
    glVertex2d(180,190);       //7

    glEnd();
}
```

SAMPLE I/0

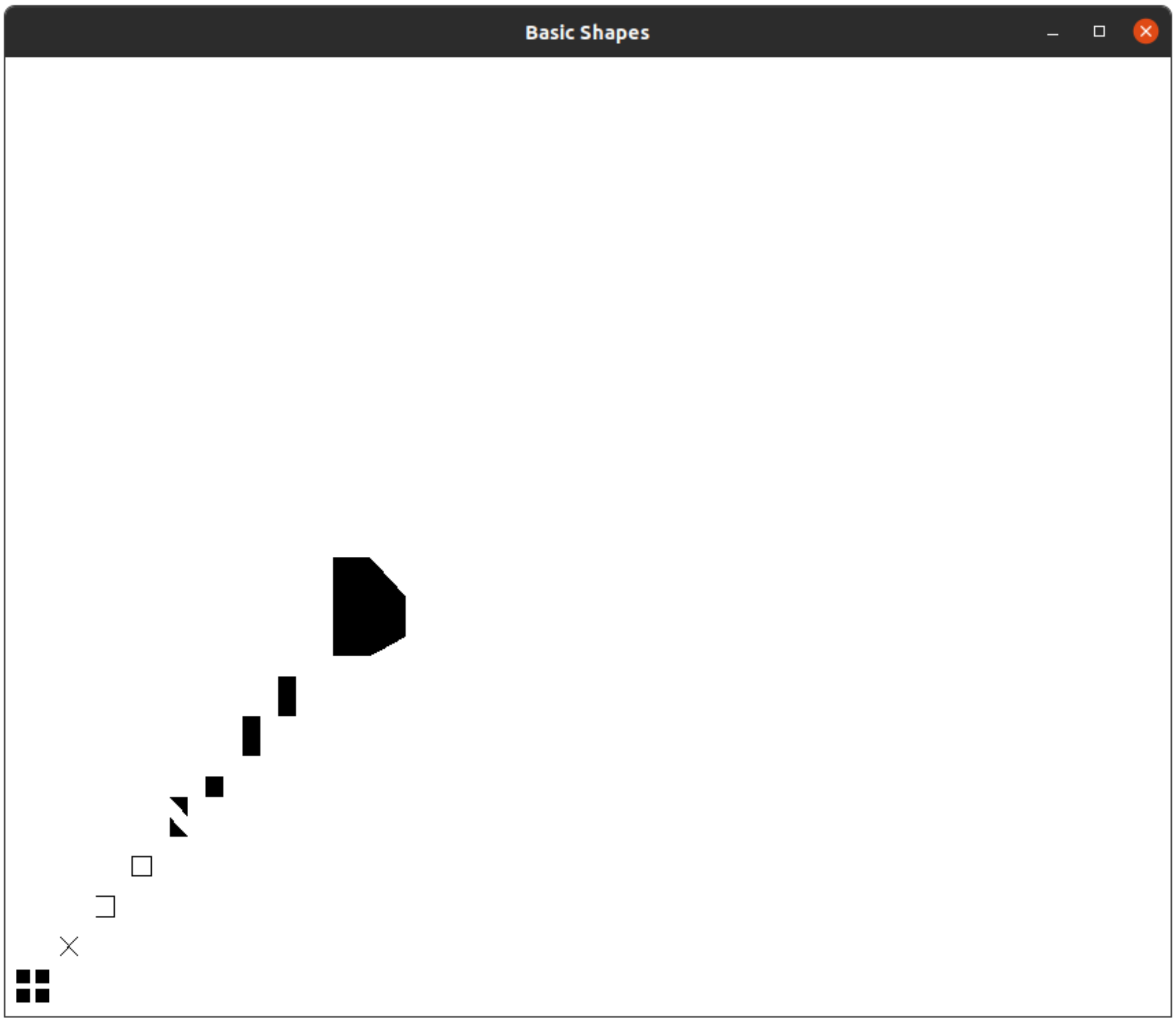

Figure 1: The primitives arranged from bottom-left to top-right

PROGRAM - 02

Drawing a Checkerboard

```
// Q: To create an output window and draw a checkerboard using OpenGL.

#include<GL/glut.h>

const int WINDOW_WIDTH = 850;
const int WINDOW_HEIGHT = 700;

void myInit();
void myDisplay();

void printCheckerBoard();
void printBoardBorder(int start_X, int start_Y, int end_X, int end_Y);
void printSquares(int min_x, int min_y, int step);

int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
    glutCreateWindow("Checkerboard");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}

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);

    printCheckerBoard();

    glFlush();
}

void printCheckerBoard() {
    // board_length -> number of unit squares by length/ X-axis
    // board_height -> number of unit squares by height/ Y-axis
    // padding -> translates the board by (padding, padding)
    // step -> denotes pixel width and length of each square in board

    int board_length = 8;
    int board_height = 8;
    int padding = 50;
    int step = 25;

    if(padding < 10) padding = 10;

    printBoardBorder(padding, padding, padding+board_length*step, padding+board_height*step);

    for(int row=0; row<board_height; row++) {
        int start;

        if(row&1)    start = 1;
        else         start = 0;

        while(start < board_length) {
            printSquares(padding + start*step, padding + row*step, step);
            start += 2;
        }
    }

}

void printSquares(int x, int y, int step) {
    //1-2-3-4-1 Quad

    glBegin(GL_QUADS);

    glVertex2d(x,y);            //1
    glVertex2d(x+step,y);       //2
    glVertex2d(x+step,y+step);  //3
    glVertex2d(x,y+step);       //4

    glEnd();
}

void printBoardBorder(int x1, int y1, int x2, int y2) {
    //1-2-3-4-....-n-1 Lines

    glBegin(GL_LINE_LOOP);

    glVertex2d(x1,y1);    //1
    glVertex2d(x2,y1);    //2
    glVertex2d(x2,y2);    //3
    glVertex2d(x1,y2);    //4

    glEnd();
}
```

SAMPLE I/0

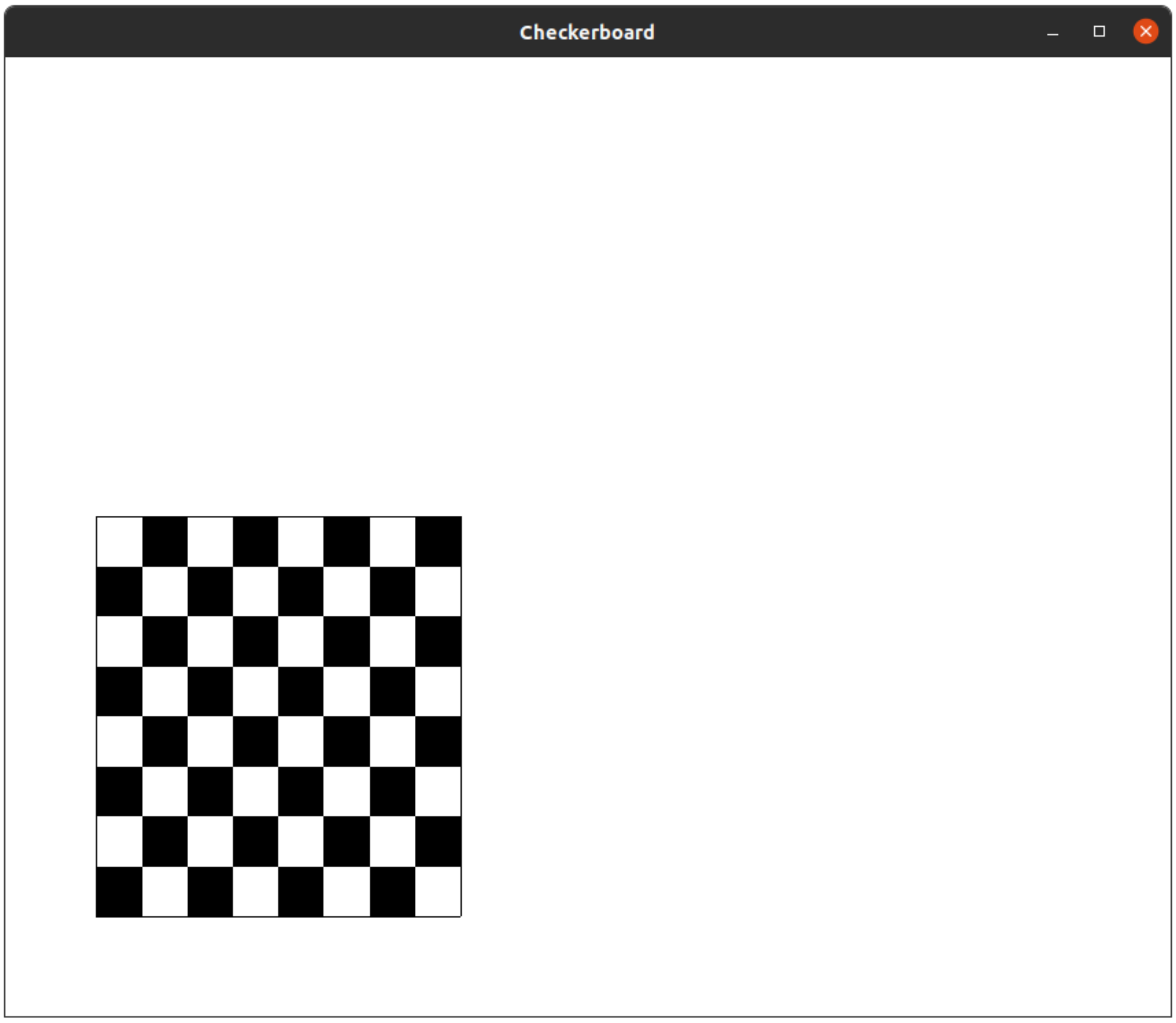

Figure 2: $8 * 8$ Checkerboard

PROGRAM - 03

Drawing a house

```c
// Q: To create an output window and draw a house using POINTS,LINES,↩
    TRIANGLES and QUADS/POLYGON.

#include<GL/glut.h>

const int WINDOW_WIDTH = 850;
const int WINDOW_HEIGHT = 700;

void myInit();
void myDisplay();

void makeBorder();
void makeDoorFrame();
void makeOuterTiles();

void printLine(int x1, int y1, int x2, int y2);
void printLineLoop(int x1, int y1, int x2, int y2);
void printTriangle(int x1, int y1, int x2, int y2, int x3, int y3);
void printQuad(int x1, int y1, int x2, int y2);

int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
    glutCreateWindow("Basic Shapes");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}

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);

    makeBorder();
    makeDoorFrame();
    makeOuterTiles();

    glFlush();
}

void makeBorder() {
    printTriangle(35,150, 115,150, 75,180);
    printLineLoop(50,100, 100,150);

    printTriangle(135,150, 215,150, 175,180);
    printLineLoop(150,100, 200,150);
}

void makeDoorFrame() {
    printQuad(62,100, 65,122);
    printQuad(85,100, 88,122);
    printQuad(62,120, 88,122);

    printQuad(162,100, 165,122);
    printQuad(185,100, 188,122);
    printQuad(162,120, 188,122);
}

void makeOuterTiles() {
    printLineLoop(62,85,88,95);
    printQuad(62,70,88,80);
    printLineLoop(62,55,88,65);
    printQuad(50,40,71,50);
    printQuad(79,40,100,50);

    printLineLoop(162,85,188,95);
    printQuad(162,70,188,80);
    printLineLoop(162,55,188,65);
    printQuad(150,40,171,50);
    printQuad(179,40,200,50);

    printQuad(105,40,122,50);
    printQuad(128,40,145,50);
}

void printLine(int x1, int y1, int x2, int y2) {
    //1-2 Line

    glBegin(GL_LINES);

    glVertex2d(x1,y1);    //1
    glVertex2d(x2,y2);    //2

    glEnd();
}

void printLineLoop(int x1, int y1, int x2, int y2) {
    //1-2-3-4-....-n-1 Lines

    glBegin(GL_LINE_LOOP);

    glVertex2d(x1,y1);   //1
    glVertex2d(x2,y1);   //2
    glVertex2d(x2,y2);   //3
    glVertex2d(x1,y2);   //4

    glEnd();
}

void printTriangle(int x1, int y1, int x2, int y2, int x3, int y3) {
    //1-2-3-1 Triangle

    glBegin(GL_TRIANGLES);

    glVertex2d(x1,y1);    //1
    glVertex2d(x2,y2);   //2
    glVertex2d(x3,y3);   //3

    glEnd();
}

void printQuad(int x1, int y1, int x2, int y2) {
    //1-2-3-4 Quad

    glBegin(GL_QUADS);

    glVertex2d(x1,y1);   //1
    glVertex2d(x2,y1);   //2
    glVertex2d(x2,y2);   //3
    glVertex2d(x1,y2);   //4

    glEnd();
}
```

SAMPLE I/0

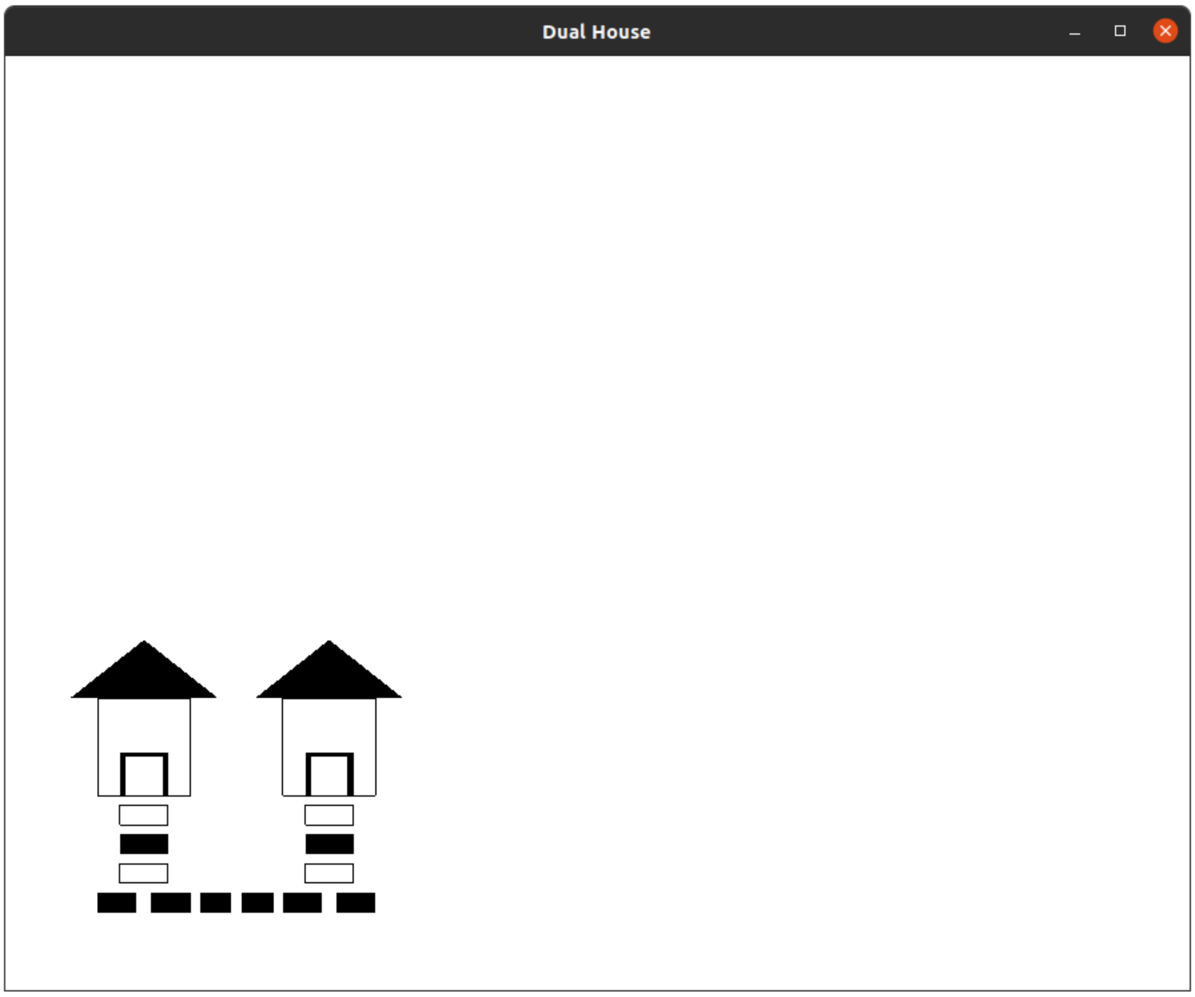

Figure 3: Two Houses connected via common footpath

RESULT

The code for studying primitive outputs were written and the outputs were verified.