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

```
1  // Q: To create an output window using OPENGL and to draw the following ↩
       basic output primitives
2  //    - POINTS
3  //    - LINES
4  //    - LINE_STRIP
5  //    - LINE_LOOP
6  //    - TRIANGLES
7  //    - QUADS
8  //    - QUAD_STRIP
9  //    - POLYGON.
10
11 // Reference: https://docs.microsoft.com/en-us/windows/win32/opengl/↩
       glbegin
12
13 #include<GL/glut.h>
14
15 const int WINDOW_WIDTH = 850;
16 const int WINDOW_HEIGHT = 700;
17
18 void myInit();
19 void myDisplay();
20
21 void printPoints();
22 void printLines();
23 void printLineStrip();
24 void printLineLoop();
25 void printTriangles();
26 void printQuads();
27 void printQuadStrip();
28 void printPolygon();
29
30 int main(int argc,char* argv[]) {
31     glutInit(&argc,argv);
32     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
33     glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
34     glutCreateWindow("Basic Shapes");
35     glutDisplayFunc(myDisplay);
36     myInit();
37     glutMainLoop();
38     return 1;
39 }
40
41 void myInit() {
```

```
42      glClearColor(1.0,1.0,1.0,0.0);
43      glColor3f(0.0f,0.0f,0.0f);
44      glPointSize(10);
45      glMatrixMode(GL_PROJECTION);
46      glLoadIdentity();
47      gluOrtho2D(0.0,640.0,0.0,480.0);
48  }
49
50  void myDisplay() {
51      glClear(GL_COLOR_BUFFER_BIT);
52
53      printPoints();
54      printLines();
55      printLineStrip();
56      printLineLoop();
57      printTriangles();
58      printQuads();
59      printQuadStrip();
60      printPolygon();
61
62      glFlush();
63  }
64
65  void printPoints() {
66      //1 - pt1, 2 - pt2, 3 - pt3, ...
67
68      glBegin(GL_POINTS);
69
70      glVertex2d(10,10);    //1
71      glVertex2d(20,20);    //2
72      glVertex2d(20,10);    //3
73      glVertex2d(10,20);    //4
74
75      glEnd();
76  }
77
78  void printLines() {
79      //1-2 Line
80
81      glBegin(GL_LINES);
82
83      glVertex2d(30,30);    //1
84      glVertex2d(40,40);    //2
85
86      glVertex2d(30,40);
87      glVertex2d(40,30);
88
```

```
89      glEnd();
90  }
91
92  void printLineStrip() {
93      //1-2-3-4-..-(n-1)-n Lines
94
95      glBegin(GL_LINE_STRIP);
96
97      glVertex2d(50,50);    //1
98      glVertex2d(60,50);    //2
99      glVertex2d(60,60);    //3
100     glVertex2d(50,60);    //4
101
102     glEnd();
103 }
104
105 void printLineLoop() {
106     //1-2-3-4-....-n-1 Lines
107
108     glBegin(GL_LINE_LOOP);
109
110     glVertex2d(70,70);   //1
111     glVertex2d(80,70);   //2
112     glVertex2d(80,80);   //3
113     glVertex2d(70,80);   //4
114
115     glEnd();
116 }
117
118 void printTriangles() {
119     //1-2-3-1 Triangle
120
121     glBegin(GL_TRIANGLES);
122
123     glVertex2d(90,90);    //1
124     glVertex2d(90,100);   //2
125     glVertex2d(100,90);   //3
126
127     glVertex2d(100,110);
128     glVertex2d(90,110);
129     glVertex2d(100,100);
130
131     glEnd();
132 }
133
134 void printQuads() {
135     //1-2-3-4-1 Quad
```

```
136
137     glBegin(GL_QUADS);
138
139     glVertex2d(110,110);  //1
140     glVertex2d(120,110);  //2
141     glVertex2d(120,120);  //3
142     glVertex2d(110,120);  //4
143
144     glVertex2d(130,130);
145     glVertex2d(140,130);
146     glVertex2d(140,150);
147     glVertex2d(130,150);
148
149     glEnd();
150 }
151
152 void printQuadStrip() {
153     //1-2-4-3 Quad1
154     //3-4-6-5 Quad2
155
156     glBegin(GL_QUAD_STRIP);
157
158     glVertex2d(150,150);       //1
159     glVertex2d(160,150);       //2
160
161     glVertex2d(150,160);       //3
162     glVertex2d(160,160);       //4
163
164     glVertex2d(150,170);       //5
165     glVertex2d(160,170);       //6
166
167     glEnd();
168 }
169
170 void printPolygon() {
171     //(1,2,3,4,..,n) Convex polygon
172
173     glBegin(GL_POLYGON);
174
175     glVertex2d(180,180);       //1
176     glVertex2d(200,180);       //2
177     glVertex2d(220,190);       //3
178     glVertex2d(220,210);       //4
179     glVertex2d(200,230);       //5
180     glVertex2d(180,230);       //6
181     glVertex2d(180,190);       //7
182
```

```
183     glEnd();
184 }
```

SAMPLE I/0

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

PROGRAM - 02

Drawing a Checkerboard

```
1  // Q: To create an output window and draw a checkerboard using OpenGL.
2
3  #include<GL/glut.h>
4
5  const int WINDOW_WIDTH = 850;
6  const int WINDOW_HEIGHT = 700;
7
8  void myInit();
9  void myDisplay();
10
11 void printCheckerBoard();
12 void printBoardBorder(int start_X, int start_Y, int end_X, int end_Y);
13 void printSquares(int min_x, int min_y, int step);
14
15 int main(int argc,char* argv[]) {
16     glutInit(&argc,argv);
17     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
18     glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
19     glutCreateWindow("Basic Shapes");
20     glutDisplayFunc(myDisplay);
21     myInit();
22     glutMainLoop();
23     return 1;
24 }
25
26 void myInit() {
27     glClearColor(1.0,1.0,1.0,0.0);
28     glColor3f(0.0f,0.0f,0.0f);
29     glPointSize(10);
30     glMatrixMode(GL_PROJECTION);
31     glLoadIdentity();
32     gluOrtho2D(0.0,640.0,0.0,480.0);
33 }
34
35 void myDisplay() {
36     glClear(GL_COLOR_BUFFER_BIT);
37
38     printCheckerBoard();
39
40     glFlush();
41 }
42
43 void printCheckerBoard() {
```

```
44      // board_length -> number of unit squares by length/ X-axis
45      // board_height -> number of unit squares by height/ Y-axis
46      // padding -> translates the board by (padding, padding)
47      // step -> denotes pixel width and length of each square in board
48
49      int board_length = 8;
50      int board_height = 8;
51      int padding = 50;
52      int step = 25;
53
54      if(padding < 10) padding = 10;
55
56      printBoardBorder(padding, padding, padding+board_length*step, padding+↩
            board_height*step);
57
58      for(int row=0; row<board_height; row++) {
59          int start;
60
61          if(row&1)   start = 1;
62          else        start = 0;
63
64          while(start < board_height) {
65              printSquares(padding + start*step, padding + row*step, step);
66              start += 2;
67          }
68      }
69
70  }
71
72  void printSquares(int x, int y, int step) {
73      //1-2-3-4-1 Quad
74
75      glBegin(GL_QUADS);
76
77      glVertex2d(x,y);            //1
78      glVertex2d(x+step,y);       //2
79      glVertex2d(x+step,y+step);  //3
80      glVertex2d(x,y+step);       //4
81
82      glEnd();
83  }
84
85  void printBoardBorder(int x1, int y1, int x2, int y2) {
86      //1-2-3-4-....-n-1 Lines
87
88      glBegin(GL_LINE_LOOP);
89
```

```
90      glVertex2d(x1,y1);    //1
91      glVertex2d(x2,y1);    //2
92      glVertex2d(x2,y2);    //3
93      glVertex2d(x1,y2);    //4
94
95      glEnd();
96  }
```

SAMPLE I/0

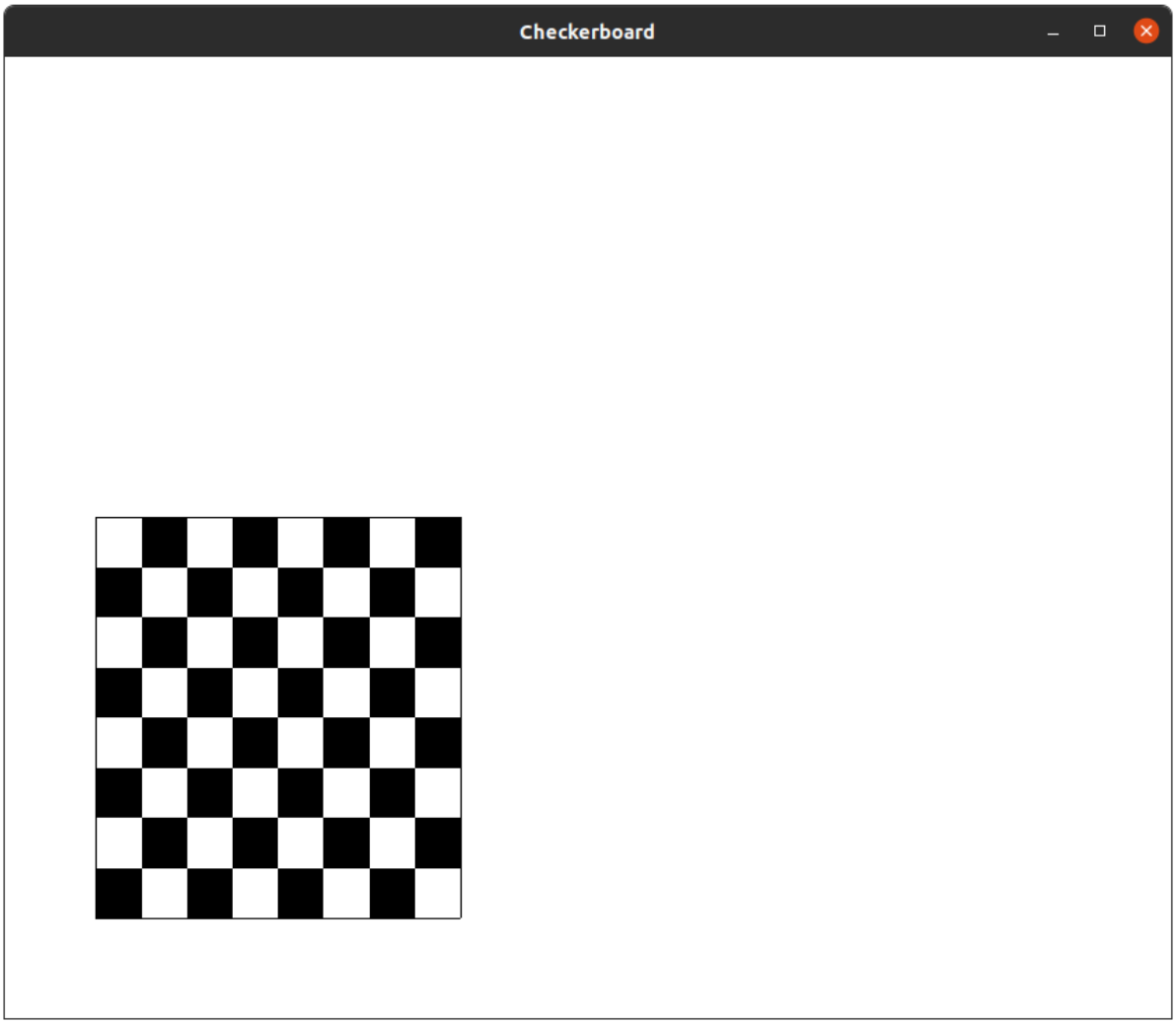

Figure 2: $8 * 8$ Checkerboard

PROGRAM - 03

Drawing a house

```
1  // Q: To create an output window and draw a house using POINTS,LINES,↩
       TRIANGLES and QUADS/POLYGON.
2
3  #include<GL/glut.h>
4
5  const int WINDOW_WIDTH = 850;
6  const int WINDOW_HEIGHT = 700;
7
8  void myInit();
9  void myDisplay();
10
11 void makeBorder();
12 void makeDoorFrame();
13 void makeOuterTiles();
14
15 void printLine(int x1, int y1, int x2, int y2);
16 void printLineLoop(int x1, int y1, int x2, int y2);
17 void printTriangle(int x1, int y1, int x2, int y2, int x3, int y3);
18 void printQuad(int x1, int y1, int x2, int y2);
19
20 int main(int argc,char* argv[]) {
21     glutInit(&argc,argv);
22     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
23     glutInitWindowSize(WINDOW_WIDTH,WINDOW_HEIGHT);
24     glutCreateWindow("Basic Shapes");
25     glutDisplayFunc(myDisplay);
26     myInit();
27     glutMainLoop();
28     return 1;
29 }
30
31 void myInit() {
32     glClearColor(1.0,1.0,1.0,0.0);
33     glColor3f(0.0f,0.0f,0.0f);
34     glPointSize(10);
35     glMatrixMode(GL_PROJECTION);
36     glLoadIdentity();
37     gluOrtho2D(0.0,640.0,0.0,480.0);
38 }
39
40 void myDisplay() {
41     glClear(GL_COLOR_BUFFER_BIT);
42
```

```
43      makeBorder();
44      makeDoorFrame();
45      makeOuterTiles();
46
47      glFlush();
48  }
49
50  void makeBorder() {
51      printTriangle(35,150, 115,150, 75,180);
52      printLineLoop(50,100, 100,150);
53
54      printTriangle(135,150, 215,150, 175,180);
55      printLineLoop(150,100, 200,150);
56  }
57
58  void makeDoorFrame() {
59      printQuad(62,100, 65,122);
60      printQuad(85,100, 88,122);
61      printQuad(62,120, 88,122);
62
63      printQuad(162,100, 165,122);
64      printQuad(185,100, 188,122);
65      printQuad(162,120, 188,122);
66  }
67
68  void makeOuterTiles() {
69      printLineLoop(62,85,88,95);
70      printQuad(62,70,88,80);
71      printLineLoop(62,55,88,65);
72      printQuad(50,40,71,50);
73      printQuad(79,40,100,50);
74
75      printLineLoop(162,85,188,95);
76      printQuad(162,70,188,80);
77      printLineLoop(162,55,188,65);
78      printQuad(150,40,171,50);
79      printQuad(179,40,200,50);
80
81      printQuad(105,40,122,50);
82      printQuad(128,40,145,50);
83  }
84
85  void printLine(int x1, int y1, int x2, int y2) {
86      //1-2 Line
87
88      glBegin(GL_LINES);
89
```

```
90      glVertex2d(x1,y1);    //1
91      glVertex2d(x2,y2);    //2
92
93      glEnd();
94  }
95
96  void printLineLoop(int x1, int y1, int x2, int y2) {
97      //1-2-3-4-....-n-1 Lines
98
99      glBegin(GL_LINE_LOOP);
100
101     glVertex2d(x1,y1);   //1
102     glVertex2d(x2,y1);   //2
103     glVertex2d(x2,y2);   //3
104     glVertex2d(x1,y2);   //4
105
106     glEnd();
107 }
108
109 void printTriangle(int x1, int y1, int x2, int y2, int x3, int y3) {
110     //1-2-3-1 Triangle
111
112     glBegin(GL_TRIANGLES);
113
114     glVertex2d(x1,y1);    //1
115     glVertex2d(x2,y2);   //2
116     glVertex2d(x3,y3);   //3
117
118     glEnd();
119 }
120
121 void printQuad(int x1, int y1, int x2, int y2) {
122     //1-2-3-4 Quad
123
124     glBegin(GL_QUADS);
125
126     glVertex2d(x1,y1);   //1
127     glVertex2d(x2,y1);   //2
128     glVertex2d(x2,y2);   //3
129     glVertex2d(x1,y2);   //4
130
131     glEnd();
132 }
```

SAMPLE I/0

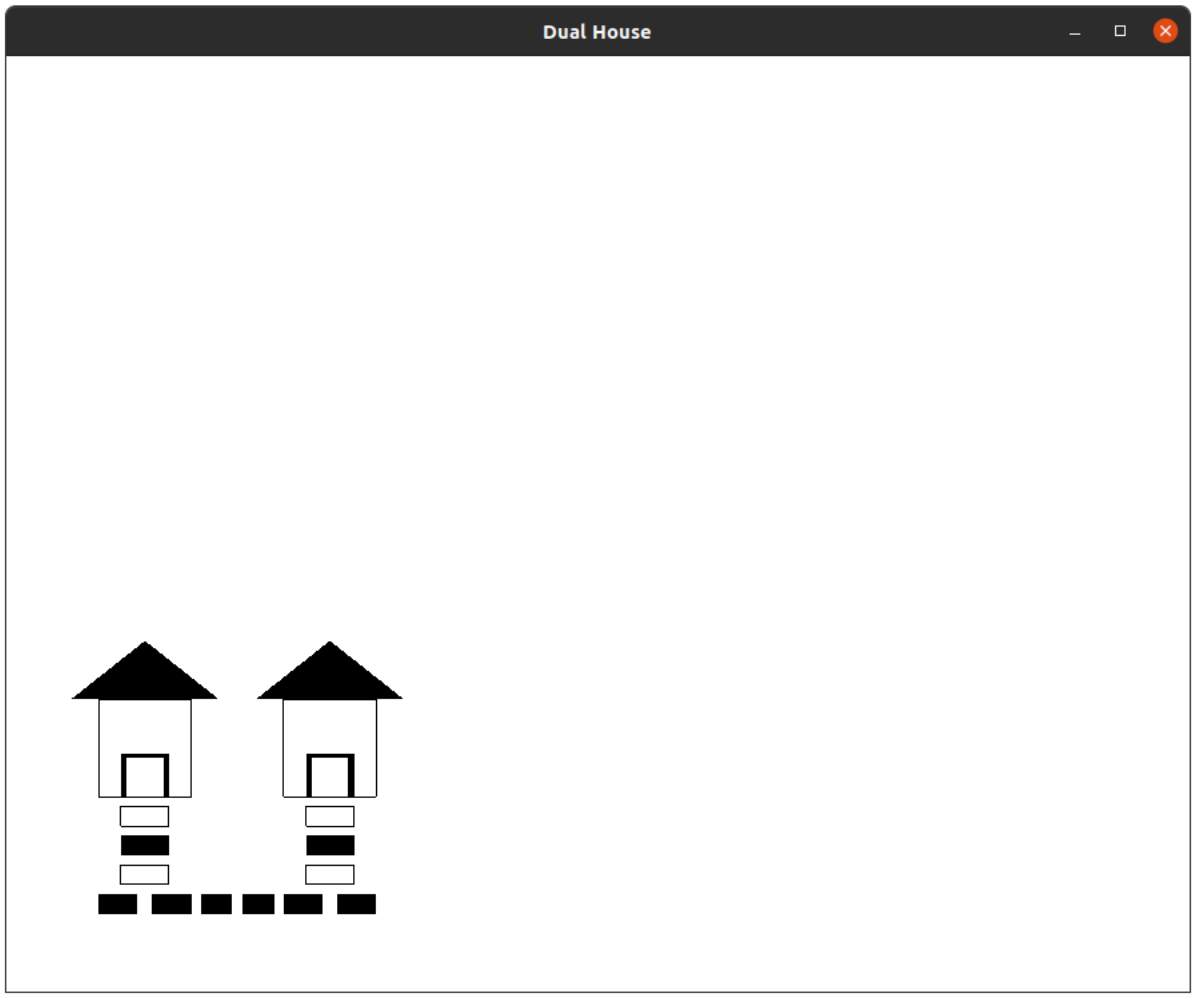

Figure 3: Two Houses connected via common footpath

RESULT

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