

Компьютер графикийн үндэс

Лаборатори №3

glViewport(), glOrtho() функцүүдийн тусламжтай өгөгдсөн зурагнаас шинэ хээ зурах

```
1  #include<windows.h>
2  #include<GL/freeglut.h>
3  #include<math.h>
4
5  float angle = 0.0f;
6  float r = 0.0, g = 0.0, b = 0.0;
7  int n = 6;
8  int rows = 4, cols = 5;
9  | int rows
10 void resize(int width, int height)
11 {
12     glViewport(0, 0, width, height);
13     glMatrixMode(GL_PROJECTION);
14     glLoadIdentity();
15     gluOrtho2D(-0.5, 0.5, -0.5, 0.5);
16     glMatrixMode(GL_MODELVIEW);
17     glLoadIdentity();
18 }
19
20 void drawHexagon(float scale)
21 {
22     glBegin(GL_LINE_LOOP);
23     |
24     for (int i = 0; i < n; i++) {
25         float a = 2.0 * 3.1415926f * float(i) / float(n);
26         float x = cosf(a) * scale;
27         float y = sinf(a) * scale;
28         glVertex2f(x, y);
29     }
30     glEnd();
31 }
32
33 void renderHexagons()
34 {
35     glColor3f(r, g, b);
36     |
37     for (int i = 0; i < 30; i++) {
38         glPushMatrix();
39         float scale = 1.0f - (float)i / 30;
40         glScalef(scale, scale, 1.0);
41         glRotatef(angle + i * 10, 0.0, 0.0, 1.0);
42         drawHexagon(1.0f);
43         glPopMatrix();
44     }
45 }
46
47 void hexagon(void)
48 {
49     glClear(GL_COLOR_BUFFER_BIT);
50
51     int windowHeight = glutGet(GLUT_WINDOW_HEIGHT);
52     int viewportWidth = windowHeight / cols;
53     int viewportHeight = windowHeight / rows;
54
55     for (int i = 0; i < rows; i++) {
56         for (int j = 0; j < cols; j++) {
57             glViewport(j * viewportWidth, i * viewportHeight, viewportWidth, viewportHeight);
58         }
59     }
60 }
```

```

59         glMatrixMode(GL_PROJECTION);
60         glLoadIdentity();
61         gluOrtho2D(-0.5, 0.5, -0.5, 0.5);
62
63
64         glMatrixMode(GL_MODELVIEW);
65         glLoadIdentity();
66         if (j%2 == 0) {
67             glScalef(-1.0, 1.0, 1.0);
68         }
69         if(i%2==0){
70             glScalef(1.0, -1.0, 1.0);
71         }
72
73         renderHexagons();
74     }
75 }
76
77 glFlush();
78 }

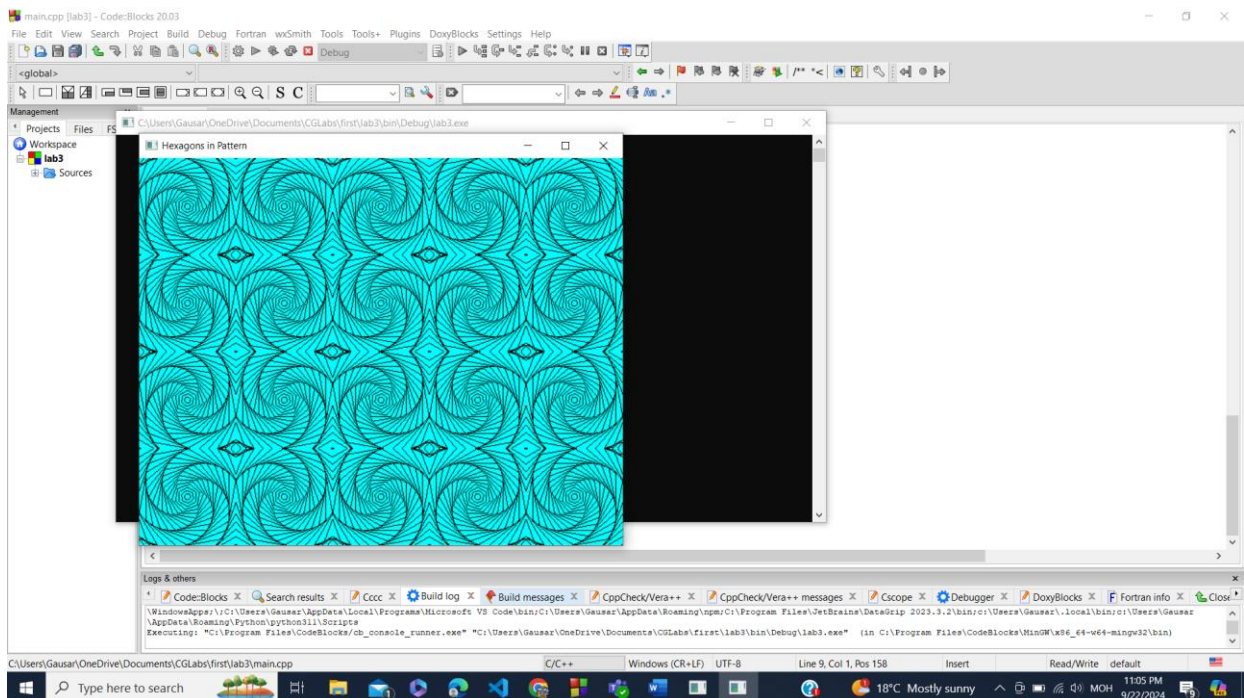
```

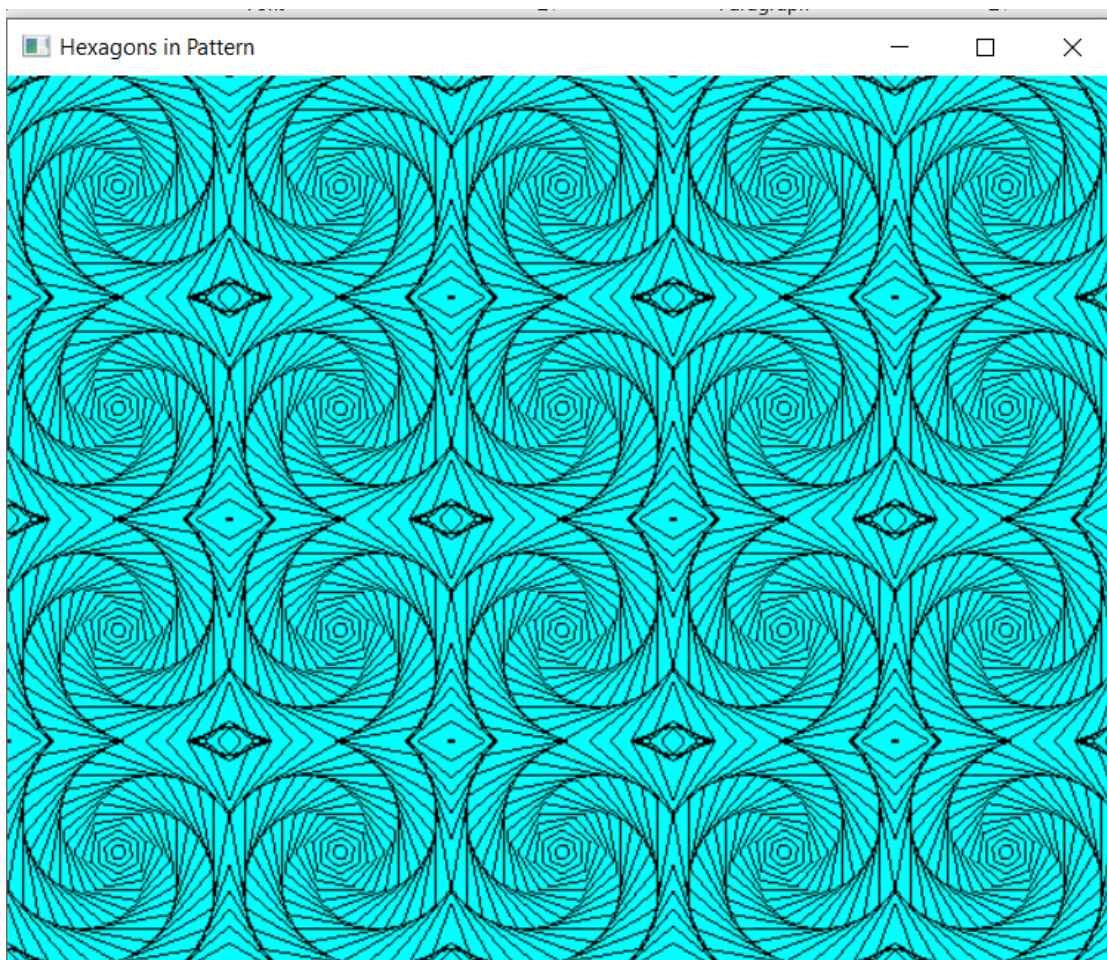
```

90 int main(int argc, char *argv[]) {
91     glutInit(&argc, argv);
92     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
93     glutInitWindowSize(600, 480);
94
95     glutCreateWindow("Hexagons in Pattern");
96
97     glutReshapeFunc(resize);
98     glutDisplayFunc(hexagon);
99
100     glClearColor(0, 1, 1, 1);
101
102     glutMainLoop();
103
104     return EXIT_SUCCESS;
105 }

```

Үр дүн:





Хэрхэн хэрэгжүүлсэн талаар:

Эхлээд `drawHexagon()` функцээр ганц 6-н өнцөгт зурсан.

`renderHexagons()` функцаар олон зургаан өнцөгтийг давхарлаж зурахын тулд хэмжээг нь жижигрүүлэх шаардлага гарсан мөн өнцөгийн эргэлт хэрэгтэй байсан. Хэмжээг нь (`scale`) хувьсагчаар дамжуулж тааруулсан. Эргэлтийг `glScalef` функцийг ашиглан `x` болон `y` тэнхлэгийн хувьд зэрэг өөрчлөлт хийсэн. Дараагаар нь өнцгийг нь 10аар нэмэгдүүлж эргэлт хийсэн.

`Hexagon` функцээр цонхоо 4x5 харьцаатай болгож хэмжээг нь тааруулж өгсөн ба `glScalef()` функцээр `viewport`-оо эргүүлэлт хийж өгсөн, ингэхдээ хэрвээ мөрний тоо тэгш байх юм бол `y` тэнхлэгийн хувьд эсрэгээрээ баганы тоо тэгш бол `x` тэнхлэгийн хувьд эргүүлэлт хийгдэж байгаа. Өөрөөр хэлбэл тэгш хэм үүсгэж харагдаж байна.