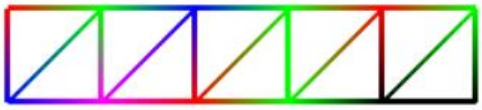
1. Am adaugat varfurile, culorile si indicii pentru a putea desena:

```
38      // coordonatele varfurilor
39      static const GLfloat vf_pos[] =
40      {
41      -25.0f, 5.0f, 0.0f, 1.0f,
42      -15.0f, 5.0f, 0.0f, 1.0f,
43      -5.0f, 5.0f, 0.0f, 1.0f,
44      5.0f, 5.0f, 0.0f, 1.0f,
45      15.0f, 5.0f, 0.0f, 1.0f,
46      25.0f, 5.0f, 0.0f, 1.0f,
47      -25.0f, -5.0f, 0.0f, 1.0f,
48      -15.0f, -5.0f, 0.0f, 1.0f,
49      -5.0f, -5.0f, 0.0f, 1.0f,
50      5.0f, -5.0f, 0.0f, 1.0f,
51      15.0f, -5.0f, 0.0f, 1.0f,
52      25.0f, -5.0f, 0.0f, 1.0f,
53
54      };
55      // culorile varfurilor
56      static const GLfloat vf_col[] =
57      {
58      1.0f, 0.0f, 0.0f, 1.0f,
59      0.0f, 1.0f, 0.0f, 1.0f,
```

```
55      // culorile varfurilor
56      static const GLfloat vf_col[] =
57      {
58      1.0f, 0.0f, 0.0f, 1.0f,
59      0.0f, 1.0f, 0.0f, 1.0f,
60      0.0f, 0.0f, 1.0f, 1.0f,
61      0.0f, 1.0f, 0.0f, 1.0f,
62      1.0f, 0.0f, 0.0f, 1.0f,
63      0.0f, 1.0f, 0.0f, 1.0f,
64      0.0f, 0.0f, 1.0f, 1.0f,
65      1.0f, 0.0f, 1.0f, 1.0f,
66      1.0f, 0.0f, 0.0f, 1.0f,
67      0.0f, 1.0f, 0.0f, 1.0f
68      };
69      // indici pentru trasarea unui triunghi
70      static const GLuint vf_ind[] =
71      {
72      6, 7, 1,0, 6, 1,
73      7, 8, 2, 1, 7, 2,
74      8, 9, 3, 2, 8, 3,
75      9, 10, 4, 3, 9, 4,
76      10, 11, 5, 4, 10,5
77
```

```
152     //glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, (void*)(0));
153     glLineWidth(3.0);
154     glDrawElements(GL_LINE_STRIP, 30, GL_UNSIGNED_INT, (void*)(0));
155
```



1.b Exercitiul suplimentar, doua cercuri unul cu raza r=10 si unul cu raza R=20, nr = 8 varfuri;

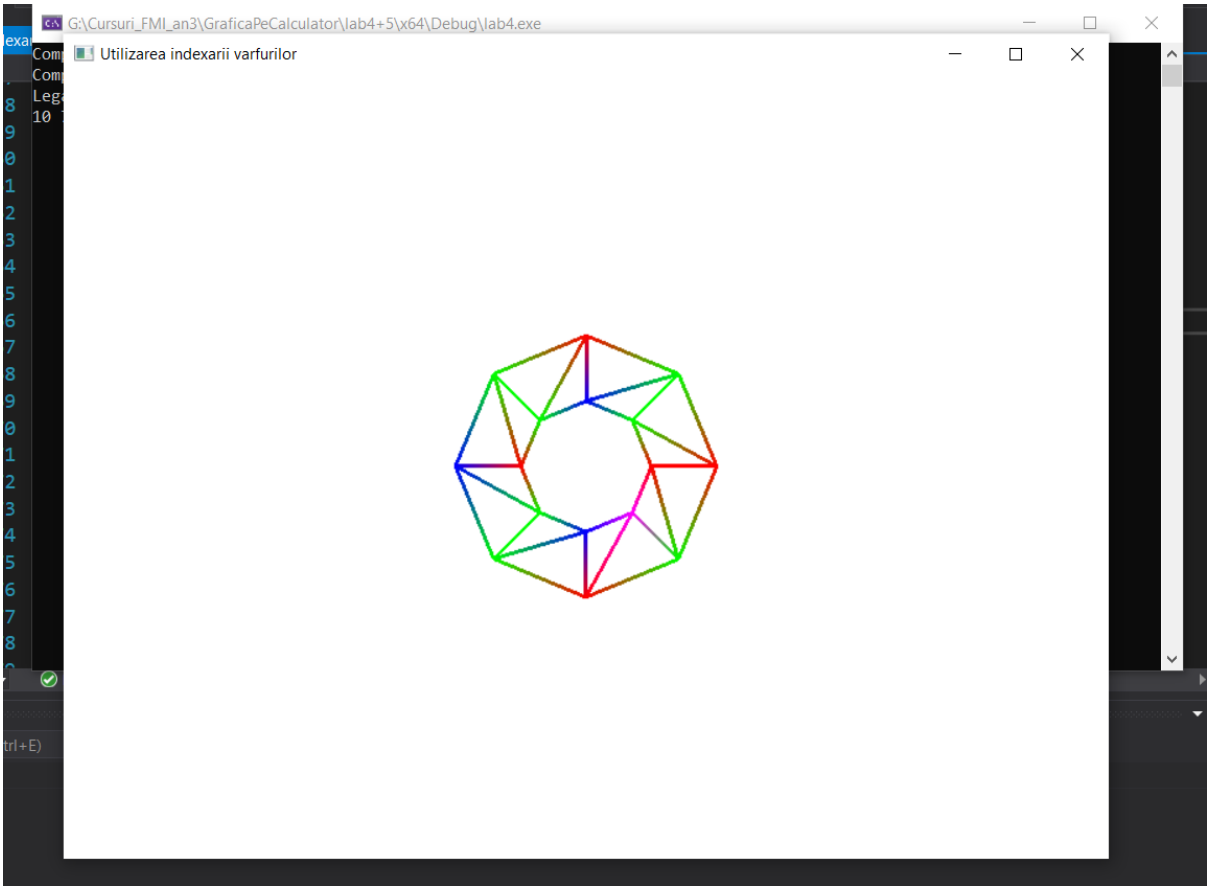
```
const float TWO_PI = 6.28;
int k, n = 8, poz=0, r=10, R=20;
```

```
42      // coordonatele varfurilor
43      GLfloat vf_pos[64]; // 16 puncte, cate 4 coord pt fiecare
44      for (k = 0; k < n; k++) {
45          theta = TWO_PI * k / n;
46          vf_pos[poz++] = r *cos(theta);
47          cout << vf_pos[poz - 1] << " ";
48          vf_pos[poz++] = r*sin(theta);
49          vf_pos[poz++] = 0.0f;
50          vf_pos[poz++] = 1.0f;
51      }
52      for (k = 0; k < n; k++) {
53          theta = TWO_PI * k / n;
54          vf_pos[poz++] = R * cos(theta);
55          vf_pos[poz++] = R * sin(theta);
56          vf_pos[poz++] = 0.0f;
57          vf_pos[poz++] = 1.0f;
58      }
```

```
59      // culorile varfurilor
60      static const GLfloat vf_col[] =
61      {
62      1.0f, 0.0f, 0.0f, 1.0f,
63      0.0f, 1.0f, 0.0f, 1.0f,
64      0.0f, 0.0f, 1.0f, 1.0f,
65      0.0f, 1.0f, 0.0f, 1.0f,
66      1.0f, 0.0f, 0.0f, 1.0f,
67      0.0f, 1.0f, 0.0f, 1.0f,
68      0.0f, 0.0f, 1.0f, 1.0f,
69      1.0f, 0.0f, 1.0f, 1.0f,
70      1.0f, 0.0f, 0.0f, 1.0f,
71      0.0f, 1.0f, 0.0f, 1.0f,
72        1.0f, 0.0f, 0.0f, 1.0f,
73      0.0f, 1.0f, 0.0f, 1.0f,
74      0.0f, 0.0f, 1.0f, 1.0f,
75      0.0f, 1.0f, 0.0f, 1.0f,
76      1.0f, 0.0f, 0.0f, 1.0f,
```

```
78      };
79      // indici pentru trasarea
80      static const GLuint vf_ind[] =
81      {
82       8,9,1,0,8,1,
83       9,10,2,1,9,2,
84       10,11,3,2,10,3,
85       11,12,4,3,11,4,
86       12,13,5,4,12,5,
87       13,14,6,5,13,6,
88       14,15,7,6,14,7,
89       15,8,0,7,15,0
90
91      };
```

```
166     // vom desena figura
167     glLineWidth(3.0);
168     glDrawElements(GL_LINE_STRIP, 64, GL_UNSIGNED_INT, (void*)(0));
169
```



2.Am creat matricea pentru translatie si cea pentru scalare.

```
158    // matrice translatie
159    matrTransl = glm::translate(glm::mat4(1.0f), glm::vec3(30.f, 30.f, 0.0));
160    //matrice scalare
161    matrScale = glm::scale(glm::mat4(1.0f), glm::vec3(2.0f, 0.5f, 1.0));
162
```

Pentru a folosi texturarea, adaugat array-ul vf_texture:

```
59     // indici pentru trasarea unui triunghi
60     static const GLuint vf_ind[] =
61     {
62     0, 1, 2, 0,2,3
63     };
64
65     static const GLfloat vf_texture[] =
66     {
67     0.0f, 0.0f,
68     1.0f, 0.0f,
69     1.0f, 1.0f,
70     0.0f, 1.0f
71
72     };
```

Am retinut si datele de texturare, iar atributul pt texturare l-am setat 2

```
86
87      // buffer-ul va contine atat coordonatele varfurilor, cat si datele de culoare, texturare
88      glBufferData(GL_ARRAY_BUFFER, sizeof(vf_col) + sizeof(vf_pos)+ sizeof(vf_texture), NULL, GL_STATIC_DRAW);
89      glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(vf_pos), vf_pos);
90      glBufferSubData(GL_ARRAY_BUFFER, sizeof(vf_pos), sizeof(vf_col), vf_col);
91      glBufferSubData(GL_ARRAY_BUFFER, sizeof(vf_col) + sizeof(vf_pos),  sizeof(vf_texture), vf_texture);
92
93      // buffer-ul pentru indici
94      glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, EboId);
95      glBufferData(GL_ELEMENT_ARRAY_BUFFER, sizeof(vf_ind), vf_ind, GL_STATIC_DRAW);
96
97      // se activeaza lucrul cu atribute; atributul 0 = pozitie, atributul 1 = culoare, acestea sunt indicate corect in VBO
98      // atributul 2 = texturare
99      glVertexAttribPointer(0, 4, GL_FLOAT, GL_FALSE, 0, NULL);
100     glVertexAttribPointer(1, 4, GL_FLOAT, GL_FALSE, 0, (const GLvoid*)sizeof(vf_pos));
101     glVertexAttribPointer(2, 2, GL_FLOAT, GL_FALSE, 0, (const GLvoid*)(sizeof(vf_pos) + sizeof(vf_col)));
102     glEnableVertexAttribArray(0);
103     glEnableVertexAttribArray(1);
104     glEnableVertexAttribArray(2);
105
```

Am adaugat functia LoadTexture(void):

```
122 }
123 void LoadTexture(void)
124 {
125     glGenTextures(1, &texture);
126     glBindTexture(GL_TEXTURE_2D, texture);
127
128     // glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT); SE REPETA SI PE OX
129     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
130     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
131
132     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
133     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
134     int width, height;
135     unsigned char* image = SOIL_load_image("text_smiley_face.png", &width, &height, 0, SOIL_LOAD_RGB);
136     glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, image);
137     glGenerateMipmap(GL_TEXTURE_2D);
138
139     SOIL_free_image_data(image);
140     glBindTexture(GL_TEXTURE_2D, 0);
141
```

In RenderFunction() am apelat-o, apoi am folosit variabilele uniforme `myMatrix, myTexture, codCol` pentru lucurul cu shaderele.

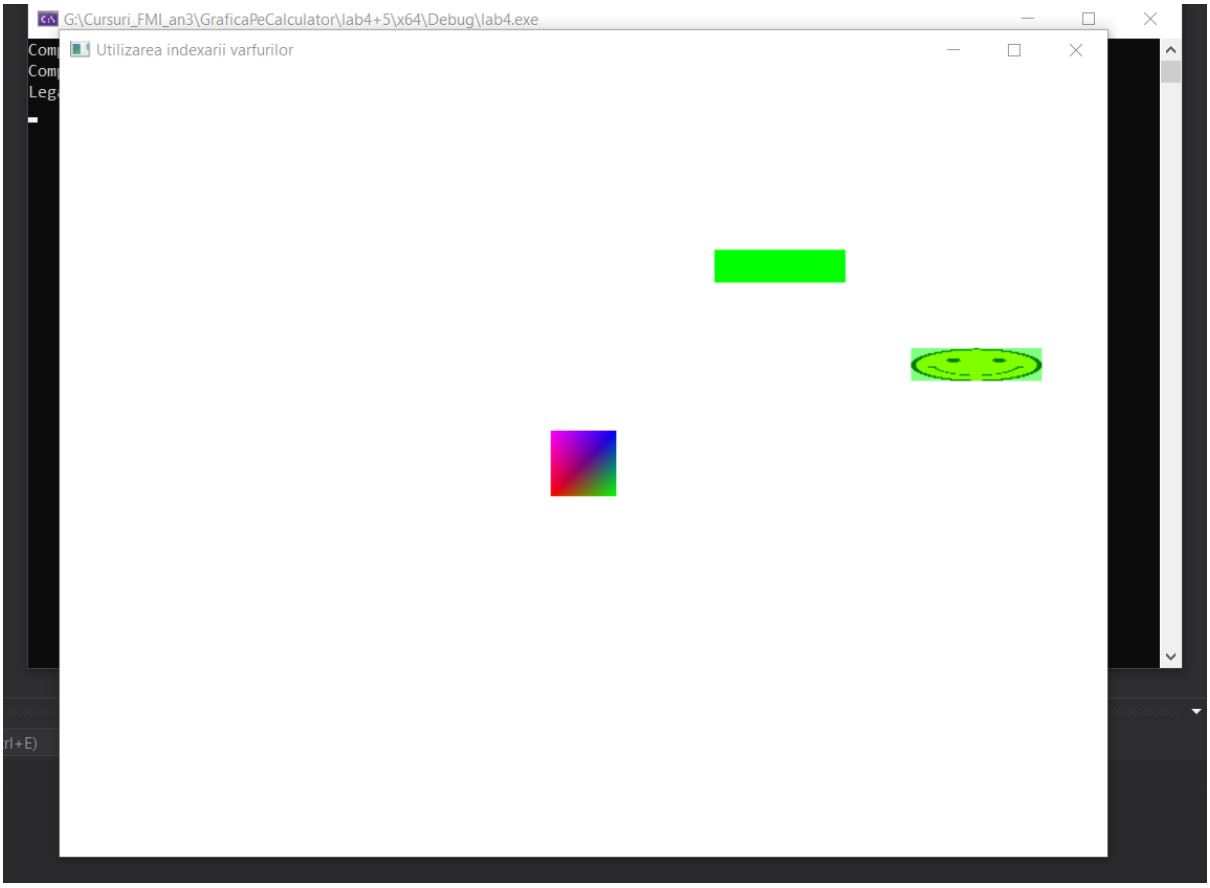
```
179     //// Desenare
180     // --- patrat initial
181     myMatrix = resizeMatrix;
182     codCol = 0; // nu va colora
183     glUniform1i(codColLocation, codCol);
184     glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0]);
185     glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, (void*)(0));
186
187     //-- patrat: aplicare scalare, apoi translatie
188     myMatrix = resizeMatrix*matrTransl *matrScale;
189     codCol = 1; // va colora cu verde
190     glUniform1i(codColLocation, codCol);
191     glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0]);
192     glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, (void*)(0));
193
194     //---patrat: aplicare translatie, apoi scalare
195     myMatrix = resizeMatrix * matrScale * matrTransl;
196     codCol = 2; // va aplica texturare
197     glUniform1i(codColLocation, codCol);
198     glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0]);
199     glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, (void*)(0));
200
```

Pentru a colora diferit figurile obtinute(sau a plica texturare) am adaugat in shaderul de varfuri

```
// Shader-ul de varfuri
 #version 400

layout(location=0) in vec4 in_Position;
layout(location=1) in vec4 in_Color;
layout(location=2) in vec2 texCoord;

out vec4 gl_Position; // comentati, daca este nevoie!
out vec4 ex_Color;
out vec2 tex_Coord;

uniform mat4 myMatrix;



void main(void)
  {
    gl_Position = myMatrix*in_Position;
    ex_Color = in_Color;
    tex_Coord = vec2(texCoord.x, 1-texCoord.y);
   }
```

Iar in shaderul de fragment:

```
in vec4 ex_Color;
in vec2 tex_Coord;
out vec4 out_Color;
vec4 green= vec4(0.0,1.0,0.0,1.0);
uniform sampler2D myTexture;
uniform int codCol; // "transmisa din main"

void main(void)
  {
   switch(codCol){
    case 0:
        out_Color = ex_Color;
        break;
    case 1:
        out_Color =green;
        break;
    case 2:
         out_Color = mix(texture(myTexture, tex_Coord), green, 0.5);
         break;
        }
  }
```


G:\Cursuri_FMI_an3\GraficaPeCalculator\lab4+5\x64\Debug\lab4.exe
Utilizarea indexarii varfurilor