

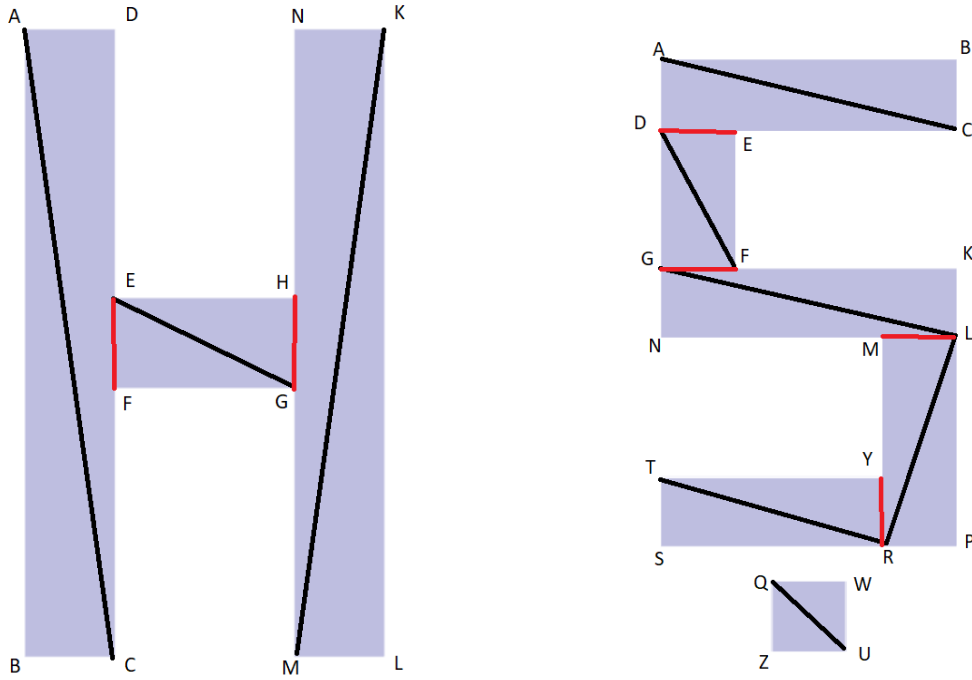
HAKAN DINDİŞ

19290234

BLM-3037 BİRİNCİ ÖDEV RAPORU

HARF ÇİZİM AŞAMALARI

Çizmem gereken harfleri önce dikdörtgenlere, daha sonra da üçgenlere ayırdım.



Devamında bu üçgenleri çizebilmek için noktaların koordinat sistemi üzerindeki gerekli (x,y) noktalarını hesapladım.

“ H ” HARFİ İÇİN ÜÇGEN NOKTA KOORDİNATLARI

ÜÇGENLER	P1	P2	P3
ABC	vec2(-0.9, 0.7)	vec2(-0.9, -0.7)	vec2(-0.7, -0.7)
ADC	vec2(-0.9, 0.7)	vec2(-0.7, 0.7)	vec2(-0.7, -0.7)
EFG	vec2(-0.7, 0.1)	vec2(-0.7, -0.1)	vec2(-0.3, -0.1)
EHG	vec2(-0.7, 0.1)	vec2(-0.3, 0.1)	vec2(-0.3, -0.1)
KNM	vec2(-0.1, 0.7)	vec2(-0.3, 0.7)	vec2(-0.3, -0.7)
KLM	vec2(-0.1, 0.7)	vec2(-0.1, -0.7)	vec2(-0.3, -0.7)

“ Ş ” HARFİ İÇİN ÜÇGEN NOKTA KOORDİNATLARI

ÜÇGENLER	P1	P2	P3
ABC	vec2(0.1, 0.7)	vec2(0.9, 0.7)	vec2(0.9, 0.5)
ADC	vec2(0.1, 0.7)	vec2(0.1, 0.5)	vec2(0.9, 0.5)
DEF	vec2(0.1, 0.5)	vec2(0.3, 0.5)	vec2(0.3, 0.1)
DGF	vec2(0.1, 0.5)	vec2(0.1, 0.1)	vec2(0.3, 0.1)
GNL	vec2(0.1, 0.1)	vec2(0.1, -0.1)	vec2(0.9, -0.1)
GKL	vec2(0.1, 0.1)	vec2(0.9, 0.1)	vec2(0.9, -0.1)
LPR	vec2(0.9, -0.1)	vec2(0.9, -0.7)	vec2(0.7, -0.7)
LMR	vec2(0.9, -0.1)	vec2(0.7, -0.1)	vec2(0.7, -0.7)
TYR	vec2(0.1, -0.5)	vec2(0.7, -0.5)	vec2(0.7, -0.7)
TSR	vec2(0.1, -0.5)	vec2(0.1, -0.7)	vec2(0.7, -0.7)
QZU	vec2(0.4, -0.8)	vec2(0.4, -1.0)	vec2(0.6, -1.0)
QWU	vec2(0.4, -0.8)	vec2(0.6, -0.8)	vec2(0.6, -1.0)

SOURCE CODE ARCHITECTURE

```

77 <script id="vertex-shader" type="x-shader/x-vertex">
78     attribute vec4 vPosition;
79     uniform float u_theta;
80     uniform vec4 u_translation;
81     uniform vec4 u_scale;
82
83     void main() {
84         gl_Position.x = ((cos(u_theta) * vPosition.x - sin(u_theta) * vPosition.y)+u_translation[0])*u_scale[0];
85         gl_Position.y = ((sin(u_theta) * vPosition.x + cos(u_theta) * vPosition.y)+u_translation[1])*u_scale[1];
86         gl_Position.z = 0.0;
87         gl_Position.w = 1.0;
88     }
89
90 </script>

```

```

92 <script id="fragment-shader" type="x-shader/x-fragment">
93     precision mediump float;
94     uniform vec4 u_color;
95     void main() {
96         gl_FragColor = u_color;
97     }
98 </script>

```

Canvas üzerinde öteleme, ölçeklendirme, döndürme ve renk değiştirme özelliklerini kontrol edebilmek için uniform değişkenler oluşturdum.

```

175 vPosition = gl.getAttributeLocation(program, "vPosition");
176 u_translation = gl.getUniformLocation(program, "u_translation");
177 u_color = gl.getUniformLocation(program, "u_color");
178 u_scale = gl.getUniformLocation(program, "u_scale");
179 u_theta = gl.getUniformLocation(program, "u_theta");
180

```

Javascript içerisinde bu değişkenlere erişmek ve manipüle edebilmek için getUniformLocation() ve getAttributeLocation() fonksiyonlarını kullandım.

```
190   theta = 0;
191   translation = [0, 0, 0, 0];
192   color = [Math.random(), Math.random(), Math.random(), 1];
193   scale = [0.75, 0.75, 0, 0];
194
```

Canvas ilk kez ekrana çizilmeden önce kullanması gereken değerleri atadım.

```
194   // Set clear color of canvas, fully opaque
195   gl.clearColor(0.0, 0.0, 0.0, 1.0);
196
197   drawScene();
```

Canvas'ın rengini değiştirdim ve drawScene() fonksiyonunu çağırarak ilk çizdirme işlemini yaptım.

```
200   function drawScene() {
201       //clear the canvas
202       gl.clear(gl.COLOR_BUFFER_BIT);
203       // turn on the attribute
204       gl.enableVertexAttribArray(vPosition);
205       //bind the position buffer
206       gl.bindBuffer(gl.ARRAY_BUFFER, buffer);
207       // Tell the attribute how to get data out of positionBuffer (ARRAY_BUFFER)
208       var size = 2; // 2 components per iteration
209       var type = gl.FLOAT; // the data is 32bit floats
210       var normalize = false; // don't normalize the data
211       var stride = 0; // 0 = move forward size * sizeof(type) each iteration to get the next position
212       var offset = 0; // start at the beginning of the buffer
213       gl.vertexAttribPointer(vPosition, size, type, normalize, stride, offset);
214
215       // set the color
216       gl.uniform4fv(u_color, color);
217       // Set the translation.
218       gl.uniform4fv(u_translation, translation);
219       //set the scale
220       gl.uniform4fv(u_scale, scale);
221       //set the rotation
222       gl.uniform1f(u_theta, theta);
223
224       // Draw the geometry.
225       var primitiveType = gl.TRIANGLES;
226       var offset = 0;
227       var count = 54;
228       gl.drawArrays(primitiveType, offset, count);
229   }
```

Bu fonksiyon vertex-shader ve fragment-shader için gerekli uniform ve attribute değişkenlerine atama işlemi yapıyor. Devamında ise canvas'ı ekrana çizdiriyor

ÖZELLİKLER VE ETKİLEŞİM TÜRLERİ

ÖZELLİK	VAR / YOK	ETKİLEŞİM TÜRÜ
Harf Çizdirme	Var	-
Renk değiştirme	Var	Mouse & GUI
Ölçeklendirme	Var	Mouse & GUI
Öteleme	Var	Keyboard (W, A , S , D)
Döndürme	Var	Mouse & GUI

RENK DEĞİŞTİRME

```
<button class="button" id="colorButton">  
  Change Color  
</button>
```

HTML dosyası içerisinde bir buton oluşturdum.

```
68   var colorButton = document.getElementById("colorButton");  
69   colorButton.addEventListener("click", function () {  
70     changeColor();  
71   });  
72
```

Javascript dosyası içerisinde oluşturduğum butona id üzerinden eriştim. Butona tıklandığında harflerin rengini değiştirmek için changeColor() fonksiyonunu çağırdım.

```
245  function changeColor() {  
246    //color = [Math.random(), Math.random(), Math.random(), 1];  
247    color = [Math.random(), Math.random(), Math.random(), Math.random()];  
248    drawScene();  
249  }  
250
```

Bu fonksiyon her çağrıldığında harflerin rengi rastlege olarak değişir.

DÖNDÜRME

```
<button class="button" id="clockwiseRotationButton">
  Rotate Clockwise
</button>
<button class="button" id="counterClockwiseRotationButton">
  Rotate Counter Clockwise
</button>
```

HTML dosyası içerisinde harfleri saat yönünde ve saat yönünün tersinde döndürebilmek için 2 tane buton tanımladım.

```
73  var clockwiseRotationButton = document.getElementById(
74    "clockwiseRotationButton"
75  );
76  clockwiseRotationButton.addEventListener("click", function () {
77    rotationDirection = "clockwise";
78    changeRotation();
79  });
80
81  var counterClockwiseRotationButton = document.getElementById(
82    "counterClockwiseRotationButton"
83  );
84  counterClockwiseRotationButton.addEventListener("click", function () {
85    rotationDirection = "counterClockwise";
86    changeRotation();
87  });
88
```

Javascript dosyası içerisinde oluşturduğum 2 butona id değerleri üzerinden eriştim. Butonlara tıklandığında harflerinin rotasyonunun değişmesi için `changeRotation()` fonksiyonunu oluşturdum.

```
251  function changeRotation() {
252    if (rotationDirection == "clockwise") {
253      theta -= 0.1;
254    } else {
255      theta += 0.1;
256    }
257
258    drawScene();
259  }
260
```

Bu fonksiyon theta değerini değiştirerek harflerin rotasyonları üzerinde değişiklik yapıyor.

ÖTELEME

```
39  window.onkeydown = function (event) {  
40      var key = String.fromCharCode(event.keyCode);  
41      switch (key) {  
42          case "W":  
43              positionDirection = "up";  
44              changePosition();  
45              break;  
46          case "A":  
47              positionDirection = "left";  
48              changePosition();  
49              break;  
50          case "S":  
51              positionDirection = "down";  
52              changePosition();  
53              break;  
54          case "D":  
55              positionDirection = "right";  
56              changePosition();  
57              break;  
58      }  
59  };  
60
```

Klavye üzerinden alınan inputlara göre harflerin gereken yöne ötelenmesini sağlayan bir switch-case bloğu oluşturdum.

```
231  function changePosition() {  
232      if (positionDirection == "right") {  
233          translation[0] += 0.07;  
234      } else if (positionDirection == "left") {  
235          translation[0] -= 0.07;  
236      } else if (positionDirection == "up") {  
237          translation[1] += 0.07;  
238      } else if (positionDirection == "down") {  
239          translation[1] -= 0.07;  
240      }  
241      drawScene();  
242  }  
243  }  
244
```

Bu fonksiyon klavyede tıklanan tuşlara göre harfleri öteliyor.

ÖLÇEKLENDİRME

```
<h1>SCALE</h1>
0<input class="slider" id="slider" type="range" min="10" max="90" step="10" value="45" /> 100
```

HTML dosyası içerisinde bir range slider oluşturdum.

```
62 document.getElementById("slider").onchange = function () {
63
64     current_value=this.value;
65
66     changeScale();
67 };
68
```

Javascript dosyası içerisinde oluşturduğum slider'a id üzerinden eriştim. Slider üzerindeki değeri current_value üzerine atadım.

```
262 function changeScale() {
263
264     scale[0] = current_value*0.01;
265     scale[1] = current_value*0.01;
266
267     drawScene();
268 }
269
```

Bu fonksiyon ise x ve y değerlerini current_value yani slider'ın o anki değeri ile çarparak harflerin ölçeklendirilmesini sağlıyor.