

BİLGİSAYAR GRAFİKLERİ ÖDEV-1

İSİM-SOYİSİM:KARDELEN DURAN

NO:19290237

Fonksiyonlar	Çalışma Durumu	Fare ve Klavye Kullanımı
Harfleri Çizdirme	Çalışıyor	-
Renk Değiştirme	Çalışıyor	Mouse
Döndürme	Çalışıyor	Mouse
Ölçeklendirme	Çalışıyor	Klavye
Öteleme	Çalışıyor	Klavye

HARFLERİ ÇİZDİRME

```
40 ///////////////////////////////////////////////////
41
42 var vertices = [
43     vec2(-0.6,0.3),vec2(-0.6,-0.3),vec2(-0.5,-0.3), //
44     vec2(-0.6,0.3),vec2(-0.5,0.3),vec2(-0.5,-0.3), //
45     vec2(-0.5,0.0),vec2(-0.2,0.3),vec2(-0.1,0.3), // K HARFİ
46     vec2(-0.5,0.0),vec2(-0.4,0.0),vec2(-0.1,0.3), //
47     vec2(-0.5,0.0),vec2(-0.2,-0.3),vec2(-0.1,-0.3), //
48     vec2(-0.5,0.0),vec2(-0.4,0.0),vec2(-0.1,-0.3), //
49     //
50     vec2(0.1,0.3),vec2(0.1,-0.3),vec2(0.2,-0.3), //
51     vec2(0.1,0.3),vec2(0.2,0.3),vec2(0.2,-0.3), //
52     vec2(0.2,0.3),vec2(0.2,0.15),vec2(0.5,-0.3), // N HARFİ
53     vec2(0.2,0.3),vec2(0.5,-0.15),vec2(0.5,-0.3), //
54     vec2(0.5,0.3),vec2(0.5,-0.3),vec2(0.6,-0.3), //
55     vec2(0.5,0.3),vec2(0.6,0.3),vec2(0.6,-0.3) //
56 ];
57
58 ///////////////////////////////////////////////////
```

K ve N harfleri üçgenlere bölünüp, bu üçgenlerin köşelerinin 2 boyutlu koordinat değerleri “vertices” adında bir Array’e atıldı

UNIFORM DEĞERLERİ.

```
71 ///////////////////////////////////////////////////
72
73 redColor = gl.getUniformLocation(program,"red");
74 red = 0.0;
75 greenColor = gl.getUniformLocation(program,"green");
76 green = 0.0;
77 blueColor = gl.getUniformLocation(program,"blue");
78 blue = 0.0;
79 transparencyValue = gl.getUniformLocation(program,"transparency");
80 transparency = 1.0;
81
82 thetaLoc = gl.getUniformLocation(program, "theta");
83 scaleLoc = gl.getUniformLocation(program,"scale");
84 translateXLoc = gl.getUniformLocation(program,"translateX");
85 translateYLoc = gl.getUniformLocation(program,"translateY");
86
87
88 ///////////////////////////////////////////////////
```

Fragment ve Vertex Shaderları içindeki uniform değerlerin lokasyonları init programı içinde çağırılıp ileride bu değerleri değiştirebilmek için değişkenlere atandı.

RENK DEĞİŞTİRME ve HIZ BUTONU FONKSİYONLARI

```
192  //////////////////////////////////////////////////
193
194      document.getElementById("redSlider").onchange = function()
195      {
196          |   red=this.value;
197      }
198      document.getElementById("greenSlider").onchange = function()
199      {
200          |   green=this.value;
201      }
202      document.getElementById("blueSlider").onchange = function()
203      {
204          |   blue=this.value;
205      }
206      document.getElementById("transparencySlider").onchange = function()
207      {
208          |   transparency=this.value;
209      }
210
211      document.getElementById("speedSlider").onchange = function()
212      {
213          |   speed=this.value;
214      }
215
216  //////////////////////////////////////////////////
```

Slider'lardan gelen değerler çekilip değişkenlere atanır.

DÖNDÜRME VE DURDURMA BUTONU FONKSİYONLARI

```
116  //////////////////////////////////////////////////
117
118      document.getElementById("rotation").onclick = function ()
119      {
120          |   rotationFlag = true;
121      }
122
123  //////////////////////////////////////////////////
124
125      document.getElementById("stop").onclick = function ()
126      {
127          |   rotationFlag = false;
128      }
129
130  //////////////////////////////////////////////////
```

Butonlar çalıştığında, global bir değişken olan rotationFlag değişkeni, döndürme ve durdurma durumlarını belirler.

YÖN BUTONLARI

```
130 //////////////////////////////////////////////////
131
132 document.getElementById("directionR").onclick = function ()
133 {
134     if(rotationFlag == true)
135     {
136         if(direction > 0)
137         {
138             direction *= -1;
139         }
140     }
141     else
142     {
143         rotate(-1);
144     }
145 }
146
147 //////////////////////////////////////////////////
148
149 document.getElementById("directionL").onclick = function ()
150 {
151     if(rotationFlag == true)
152     {
153         if(direction < 0)
154         {
155             direction *= -1;
156         }
157     }
158     else
159     {
160         rotate(1);
161     }
162 }
163
164 //////////////////////////////////////////////////
```

Harflerin hangi yöne doğru döndürüleceğini belirleyen butonlardır. rotationFlag'den gelen değere göre dönme durumu belirlenir. Buton fonksiyonları çağırıldığında:

-harfler dönüyorsa dönmenin yönünü belirler,

-harfler dönmüyorsa rotate fonksiyonunu kullanarak theta değeri kadar istenilen yöne döndürür.

```
285 function rotate(d)
286 {
287     theta += 0.1*d;
288     gl.uniform1f(thetaLoc,theta);
289 }
```

RESET BUTONU FONKSİYONU

```
164 //////////////////////////////////////////////////
165
166 document.getElementById("reset").onclick = function ()
167 {
168     resetFlag = true;
169     clearInterval(interval2);
170     rotationFlag = false;
171     cdFlag = true;
172     speed = 10;
173 }
174
175 //////////////////////////////////////////////////
```

Harflerin rengini ve bütün sayısal değerlerini sıfırlar.

ÖTELEME VE BOYUTLANDIRMA BUTONU FONKSİYONLARI

```
175 //////////////////////////////////////////////////
176
177 document.onkeydown = function(e)
178 {
179     switch (e.key) {
180         case "u":
181             case "U":
182                 scale += 1;
183                 break;
184         case "d":
185             case "D":
186                 if(scale -1 > 0)
187                 {
188                     scale -=1;
189                 }
190                 break;
191         case "ArrowLeft":
192             translateX -= 0.1;
193             break;
194         case "ArrowRight":
195             translateX += 0.1;
196             break;
197         case "ArrowUp":
198             translateY += 0.1;
199             break;
200         case "ArrowDown":
201             translateY -= 0.1;
202             break;
203     }
204 }
205
206 //////////////////////////////////////////////////
```

Klavyeden basılacak olan tuşları dinleyen fonksiyonlardır. Scale değeri artırılıp azaltılarak harflerin boyutu belirlenir; u-U tuşu harfleri büyütür, d-D tuşu harfleri küçültür.

translateX ve translateY değerleri artırılıp azaltılarak, yön tuşları ile harfler istenilen yönde ötelenir.

RANDOM COLOR GENERATOR BUTONU FONKSİYONU

```
206 //////////////////////////////////////////////////
207
208 const cd = document.getElementById("rcg");
209 cd.addEventListener("click", function ()
210 {
211     if(cdFlag == true)
212     {
213         interval2 = setInterval(() => {
214             red = getRandomNumber();
215             green = getRandomNumber();
216             blue = getRandomNumber();
217         }, 500);
218         cdFlag = false;
219     }
220 })
221
222 //////////////////////////////////////////////////
223
224 const cdstop = document.getElementById("rcgstop");
225 cdstop.addEventListener("click", function ()
226 {
227     clearInterval(interval2);
228     cdFlag = true;
229 })
230
231 //////////////////////////////////////////////////
232
```

Random Color Generator butonuna basıldığında belli aralıklarla harflere rastgele renk atayan fonksiyondur. 3 renge birden getRandomNumber() fonksiyonundan gelen 0 ile 1 arasındaki rastgele sayılar atanır.

```
278 function getRandomNumber()
279 {
280     var rm = (Math.random()*(1.0 - 0.1) + 0.1).toFixed(2);
281     console.log(rm);
282     return rm;
283 }
```

Fonksiyonun durumu cdFlag ile belirlenir. Aralıklı renk ataması “Random Color Generator”ı durdur” butonu ile durdurulabilir.

RENDER FONKSİYONU

```
238 function render()
239 {
240
241     gl.clear(gl.COLOR_BUFFER_BIT);
242
243
244     if(rotationFlag == true)
245     {
246         theta += 0.1*direction;
247         gl.uniform1f(thetaLoc, theta);
248     }
249     else
250     {
251         clearTimeout(timeOut);
252     }
253
254     if(resetFlag == true)
255     {
256         red = 0;green = 0; blue = 0;transparency = 1;
257         scale = 25; translateX = 0; translateY = 0;
258         resetFlag = false;
259         theta = 0;
260         gl.uniform1f(thetaLoc, theta);
261     }
262
263     gl.uniform1f(redColor,red);
264     gl.uniform1f(greenColor,green);
265     gl.uniform1f(blueColor,blue);
266     gl.uniform1f(transparencyValue,transparency);
267     gl.uniform1f(scaleLoc,scale/25);
268     gl.uniform1f(translateXLoc,translateX);
269     gl.uniform1f(translateYLoc,translateY);
270
271
272     gl.drawArrays(gl.TRIANGLES, 0, 36);
273
274     timeOut = setTimeout(function () {render();},speed);
275 }
```

Harflerin bütün değerlerinin güncellendiği, resetlendiği ve çizdirildiği fonksiyondur. Fonksiyon belli aralıklarla kendini çağırarak init programından gelen yeni değerleri, hesaplanması için Shader'lara atar.

VERTEX VE FRAGMENT SHADER

```
298 <script id="vertex-shader" type="x-shader/x-vertex">
299     attribute vec4 vPosition;
300     uniform float theta;
301     uniform float scale;
302     uniform float translateX, translateY;
303
304     void main() {
305         gl_Position.x = (scale * (cos(theta) * vPosition.x - sin(theta) * vPosition.y) + translateX);
306         gl_Position.y = (scale * (sin(theta) * vPosition.x + cos(theta) * vPosition.y) + translateY);
307         gl_Position.z = 0.0;
308         gl_Position.w = 1.0;
309     }
310
311 </script>
312 <script id="fragment-shader" type="x-shader/x-fragment">
313     precision mediump float;
314     uniform float red;
315     uniform float green;
316     uniform float blue;
317     uniform float transparency;
318     void main() {
319         gl_FragColor = vec4( red, green, blue, transparency );
320     }
321 </script>
```

Shaderlar, init programından gelen değerleri hesaplarlar. Güncel değerleri uniform veya attribute türü değişkenlerde tutarlar.

Vertex Shader noktaların pozisyonunu, Fragment Shader harflerin renk değerlerini hesaplar.