AD: Moahmmed Elazab

No: 030117126

MERGE

Oyuncu iki şekli aynı yere getirmeye ya da birleştirmeye çalışır, becerirse bir sonraki levele atlayarak şekillerin boyutu küçülür, ve hızı artar.

Oyunun amacı oyuncunun kafasını karıştırmak. Nedeni de bu modda oyuncu iki elini kullanarak iki farklı şekli hareket ettirmeye çalışacak, bundan çıkabilecek kafa karışıklığı oyuncu bir şekile bakarken ve hareket ettirdiğini düşünürken aslında diğer şekli hareket ettirdiğini geç fark edip kaybeder.

Yukarıda "bu modda" dememin sebebi ilerde oyunu geliştirerek Multiplayer şeklinde oynanabilir hale gelmesi, bu modda da iki kişi oynayabilir ama oyun baştan buna göre tasarlanmadığı için sayılmaz.

Bunun uygulamalı hali odaklanmanı veya dikkatini kaybetmemen gerektiren oyun kategorisine girerek iki elini farklı bir şekilde kullanman herhangi bir aktivitenin egzersizi olabilir. Örenğin, piano gibi müzik aleti çalmak.

Oyunun theme'ini kaybetmemek için FL stduio kullnarak özel müziği yapıldı.

Note;

Oyun bilgisayarımda sadece Micorsoft Edge tarayıcısını kullanarak çalışıyordu, bu her bilgisayar için farklı olabilir, çalışmaması durumunda local server oluşturan bir uygulama kullanmak gerekebilir. Xampp gibi. Kullandığımda bütün tarayıcılarda çalışmıştır.

Oyunun kodu aşağıdaki resimlerde gibi;

```
var daire=[2];
var butonlar=[];
var oklar=[]
var renkler1=['#c2bf5f','#1f7d93', '#1f936f'];
var renkler2=['#9a4022','#6d1691', '#2e3ba6'];
var circleSize=155;
var renk;
var ivme1=0.6;
var ivme2=-0.6;
var baslangicKontrol=0;
var levelKontrol=0;
var myFont;
var startSong;
var playSong;
var levelSong;
var finishSong;
var WinningSong;
var timer=10;
var timerChange=5;
var score=0;
var renk1='#0d88a5';
var renk2='#1f936f';
var renkFont='#DD255D';
```

Yukarıda kullanacağımız değişkenlerinin tanımlama işlemini yaptık.

Color Wheel' deki complementary seçeneği kullanılarak birbirine uygun renkler seçildi.

Daire boyutu da draw fonskisyonunda çıkarılacak değerle uygun olarak 155 seçildi, sonradan değiştirilebilir istenirse.

```
function preload() {
  myFont = loadFont('libraries/Fonts/StartingFont.ttf');
  startSong = loadSound("libraries/Sounds/Game-Project-8-StartingMusic.mp3");
  playSong = loadSound("libraries/Sounds/Game-Project-8-PlayingMusic.mp3");
  levelSong = loadSound("libraries/Sounds/Game-Project-8-LevelMusic.mp3");
  finishSong = loadSound("libraries/Sounds/Game-Project-8-FinishingMusic.mp3");
  WinningSong = loadSound("libraries/Sounds/Game-Project-8-WinningMusic.mp3");
  startSong.setVolume(0.1);
  playSong.setVolume(0.1);
  levelSong.setVolume(0.1);
  finishSong.setVolume(0.1);
  WinningSong.setVolume(0.1);
}
```

Yukarıdaki Preload fonksiyonunda oyunun theme'ine özgün olsun diye yaptığımız müziğin dosyaları yükleme işlemini yapıyoruz. Son üç satırda ise ses seviyesinin ayarlamasını yaptık.

```
36 function setup() {
37   createCanvas(900, 600, WEBGL);
38   background(51);
39   Giris();
40 }
```

Yukarıdaki Setup fonksiyonunda canvasın boyutlarını ve hangi modda çalışacağını belirttik. Ardından arkaplan rengini ayarladık sonra Giriş ekranını gösterecek fonksiyonu çağırdık.

WEBGL modunu seçilmesi sebebi Font'u istediğimiz şekilde kullanmak istiyorsak ancak WEBGL'de çalıştırabiliriz. Ayrıca, web için implement etmek istediğimizde sorun yaşamayız.

```
42 function Giris() {
43   startSong.loop();
44   fill(renkFont);
45   textFont(myFont);
46   textSize(20);
47   textAlign(CENTER);
48   text('Press Space to Start The Game!', 0,0);
49   text('HAVE FUN!', 0, -(height/2)+25);
50   butonlar[1]=new Buton('W', -width/4-40,height/2-150,40);
51   butonlar[1].show();
52   butonlar[2]=new Buton('A', -width/4-100,height/2-90,40);
53   butonlar[2].show();
54   butonlar[3]=new Buton('S', -width/4-40,height/2-90,40);
55   butonlar[3].show();
56   butonlar[4]=new Buton('D', -width/4+20,height/2-90,40);
57   butonlar[4].show();
58   butonlar[5]=new Buton('', width/4-40,height/2-155,40);
59   butonlar[5].show();
60   butonlar[6]=new Buton('', width/4-100,height/2-95,40);
61   butonlar[6].show();
62   butonlar[7]=new Buton('', width/4-40,height/2-95,40);
63   butonlar[7].show();
64   butonlar[8]=new Buton('', width/4+20,height/2-95,40);
65   butonlar[8].show();
66   angleMode(DEGREES);
67   oklar[1]=new Ok(width/4-40,height/2-177.5, 10);
68   oklar[1].show();
69   push();
70   translate(width/4-112.5,height/2-105);
85   rotate(90);
86   oklar[4]=new Ok(0,0, 10);
87   oklar[4].show();
88   pop();
89
90 }
```

Yukarıdaki Giriş fonksiyonunu ekranın nasıl bir formatı olacağını ayarladık.
50. Satırda en altta olan classları kullanarak oyuncunun yönlendirmesi için girş ekranında buton şekillerini oluşturduk.
58. Satırda oklar için rectlerin içi boş bırakıldı.
66. Satırdan itibaren başlayan kodlar ise okları göstermek için show, gerekirse yönünü değiştirmek translate sonra rotate + push ve pop, fonksiyonlarını kullandık.

```
92 function draw() {
93   if(baslangicKontrol==1){
94   background(51);
95   fill(renkFont);
96   textFont(myFont);
97   textSize(20);
98   text((timer-1)+'sc',  (width/2)-40,-(height/2)+20);
99   text('score '+score, -(width/2)+70,-(height/2)+20);
100  timerF();
101  daire[1].update();
102  daire[1].show( daire[2]);
103  daire[1].checkGameStatus();
104  daire[2].update();
105  daire[2].show( daire[1]);
106  daire[2].checkGameStatus();
107
108  while (daire[1].merge(daire[2])) {
109    score+=10+timer;
110    levelSong.play();
111    levelKontrol++;
112    levelAtla(circleSize-=25, levelKontrol);
113    timer=7+timerChange;
114    timerChange--;
115      }
116  }
117 }
```

93. Satırda, space'e basıldığı an baslangıcKontrol değişkeni 0dan 1e set edilerek oyunu başlatmış oluyoruz.

98. Satırda, Score ve Timerin nerede gösterileceğini buradan ayarladık.

timer-1 şeklinde göstermemizin sebebi timer sıfırlandığı anda -1 göstermemesi için -1'e değil de 0'a gelince bitimiş gibi görünsün.

Bir sonraki satırda da Timer fonksiyonunu çağırarak timer'i başlatıyoruz.

108. Satırda, iki dairenin intialladıktan sonra hareketlerini güncellemek için update, arasındaki bağlantıyı göstermek için show, ve sınırlara temas etme kontrolu için checkGameStatus fonksiyonlarını kullandık.

109. Satırda, kullanıcı kaç saniyede level atlyabilirse ona bağlı olarak kendi performansına özel bir score'u olacak - küçük bir sayı olmasın diye rastgrle bir sayıyı seçtik, bu durumda 10.

110. Satırda, kullanıcının level atladığı anda kısa bir müzik çalacak sonra levelKontrol değişkenine hangi levelde olunduğu atanacak.

113. Satırda, timerin değişmesi şu şekilde çalışır; timerin orijinal değerine reset edip sonra ona timerChange'deki olan değeri ekliyor. Eklemenin sebebi de level atladıkça zaman artacak ama oyunun zorlaştığı için bir önceki levelle aynı süre verilmesi mantıklı olmaz. Bu yüzden abartılı olmayan şekilde arttırmamız lazım.

114. Satırda, bir sonraki levelin süresine ekleyeceğimiz saniye sayısı daha az olacak.

```
122 function baslat(){
123   daire[1] = new Daire(random(-(width/2)+120,(width/2)-120),random(-(height/2)+120, (height/
124   daire[2] = new Daire(random(-(width/2)+120,(width/2)-120),random(-(height/2)+120, (height/
125   while(daire[1].merge(daire[2]) || daire[1].checkTheFirstGameStatus() || daire[2].checkTheF
126 }
127
128
129 function levelAtla(size, levelKontrol){
130   if(levelKontrol==6){
131     finishGame('Congrats!', 'You have reached the final level!', score);
132     WinningSong.loop();
133   }
134   else{
135     daire[1] = new Daire(random(-(width/2)+120,(width/2)-120),random(-(height/2)+120, (heigh
136     daire[2] = new Daire(random(-(width/2)+120,(width/2)-120),random(-(height/2)+120, (heigh
137     while(daire[1].merge(daire[2]) || daire[1].checkTheFirstGameStatus() || daire[2].checkTh
138   }
139 }
```

Yukarıdaki birbirine benzeyen iki fonksiyon oyunu başlatmamızı ve level atlamamızı sağlayan fonksiyonlardır.

Buradaki while döngüleri, daire classın içindeki daire durumunu kontrol eden fonkiyonları çağırır, gerektiği durumda oyunu yeniden/aynı levelde başlatır.

LevelAtla fonksiyonunda da oyun son duruma gelirse oyunu sonlandıracak fonksiyonu çağırır.

```
138 function timerF(){
139   if(baslangicKontrol==1){
140     if (frameCount % 60 == 0 && timer > 0) {
141      timer --;
142    }
143    if (timer == 0) {
144      fill(renkFont);
145      textFont(myFont);
146      textSize(20);
147      finishGame("TIME OUT!", "", score);
148      }
149    }
150 }
151
152 function finishGame(shownText, opText, opTextforScore){
153   fill(renkFont);
154   textFont(myFont);
155   textSize(27);
156   text(shownText, 0,(-height/2)+50);
157   textSize(15);
158   text(opText, 0,30);
159   textSize(27);
160   text('Your Score is ' + opTextforScore +' !', 0,(height/2)-100);
161   playSong.stop();
162   finishSong.loop();
163   circleSize=155;
164   ivme1=0.6;
165   ivme2=-0.6;
166   baslangicKontrol=0;
167   levelKontrol=0;
168   timer=10;
169   timerChange=5;
170   score=0;
171   text('Press Space to Play Again!', 0,(height/2)-30);
172
173 }
```

Timer fonksiyonunda şu işlem gerçekleştirilir; frameCount 60'a bölünüyorsa bir saniye geçmiş anlamına gelir dolayısıyla timer süresinden 1 azaltır, 0’a gelince de oyunu sonlandıracak fonksiyon çağırır.

finishGame fonksiyonu az önce kullandığımız oyunu sonlandıran fonksiyon.

Oyun bittiğinde göstermek istediğimiz Textleri parametre olarak atıp uygun yerde kendi yazdırıyor ekrana.

Fonksiyonun ikinci kısmında ise bütün reset edilmesi gerekn değişkenleri reset ediyoruz.

```
179 function keyPressed(){
180   if(keyCode==UP_ARROW){
181   daire[1].direction(0, ivme2-=0.5);
182   }
183   else if(keyCode==DOWN_ARROW){
184    daire[1].direction(0, ivme1+=0.5);
185   }
186   else if(keyCode==RIGHT_ARROW){
187    daire[1].direction(ivme1+=0.5, 0);
188   }
189   else if(keyCode==LEFT_ARROW){
190    daire[1].direction(ivme2-=0.5, 0);
191   }
192
193   if(keyCode==87){
194   daire[2].direction(0, ivme2-=0.5);
195   }
196   else if(keyCode==83){
197    daire[2].direction(0, ivme1+=0.5);
198   }
199   else if(keyCode==68){
200    daire[2].direction(ivme1+=0.5, 0);
201   }
202   else if(keyCode==65){
203    daire[2].direction(ivme2-=0.5, 0);
204   }
205   if(keyCode==32&&baslangicKontrol==0){
206   baslangicKontrol=1;
207   finishSong.pause();
208   startSong.pause();
209   WinningSong.pause();
210   playSong.loop();
211   baslat();
212   }
213 }
```

Objeleri hareket ettirmek için hangi butonların kullanılacağını ve ne işlemlerin yapılacağını keyPressed fonksiyonundan ayarladık.

Oyunu zorlaştırmak için herhangi bir butona basıldığı an yön değişmekle berbaber ivme artar.

201.Satırda, space basılıysa oyunu başlatır ve baslangıcKontrol 0'dan 1'e alır.

Burada space butonunu ile beraber baslangıcKontrol değişkenini logical AND işlemine sokmamızın sebebi oyuncuya kaybedene/kazanana kadar oyun ortasında oyunu yeniden başlatma hakkı sağlamamak.

203. Satırda, oyun esnasında çalacak müziği buradan döngüye sokarak başlatılıyoruz.

```
210 class Daire{
211   constructor(x,y,size, renk){
212   let xvelocity, yvelocity;
213   this.x=x;
214   this.y=y;
215   this.size=size;
216   this.renk=renk;
217   this.xvelocity=1;
218   this.yvelocity=0;
219   }
220   update(){
221   this.x = this.x + this.xvelocity;
222   this.y = this.y + this.yvelocity;
223   }
224   show(obje){
225     noFill();
226     strokeWeight(10);
227     stroke(this.renk);
228     ellipse(this.x, this.y, this.size);
229     strokeWeight(1);
230     for(let i=0; i<15; i++){
231                   circle(random(this.x,obje.x/2),random(this.y,obje.y/2),random(30));
232     }
233     for(let i=0; i<9; i++){
234                circle(this.x,this.y,random(50));
235     }
236     strokeWeight(1);
237     rectMode(CENTER);
238     for(let i=0; i<9; i++){
239               rect(this.x,this.y,random(50),random(50));
240     }
241 }
```

Yukarıda Daire calssın bir kısmını görebiliyoruz.

Constructor'a dairenin koordinatları, boyutu ve rengi atanır.

220. Satırda, üstteki draw fonksiyonun içinde hareketi gücellemek için kullandığımız update fonksiyonun işlemini yaptık.

224. Satır, objeyle ilgili bütün her şeyi görmemizi sağlayan fonksiyon.

228. Satırda, oyuncunun orijinal olarak hareket ettirdiği daire bu satırda ellipse olarak oluşturulur.

230. Satırda, dairenin içindeki şekilleri + sürekli değişen rastegele boyutlarda daireler oluşturarak iki dairenin arasındaki bağlantıyı kurmuş oluruz.

Sondaki iki for döngüsü dairenin içinde şekilleri oluşturmak için kullandık.

```
242   direction(x,y){
243    this.xvelocity=x;
244    this.yvelocity=y;
245   }
246   merge(obje){
247     var d = dist(this.x, this.y, obje.x, obje.y);
248
249     if (d <=17) {
250       noFill();
251       strokeWeight(random(7));
252       stroke(renkFont);
253       circle(this.x-20, this.y-20, random(200));
254     return true;
255     } else {
256     return false;
257     }
258   }
259   checkTheFirstGameStatus(){
260     if (
261       this.x> (width/2)-(5+this.size/2) ||
262       this.x < (-width/2)+(5+this.size/2) ||
263       this.y> (height/2)-(5+this.size/2)||
264       this.y < (-height/2)+(5+this.size/2))
265       {
266         return true;
267       }
268   }
269   checkGameStatus() {
270   if (
271     this.x> (width/2)-(5+this.size/2) ||
272     this.x < (-width/2)+(5+this.size/2) ||
273     this.y> (height/2)-(5+this.size/2)||
274     this.y < (-height/2)+(5+this.size/2)){
275     finishGame('Game Ended!','', score);
276     }
277   }
278 }
```

Yukarıda Daire classın ikinci kısmını görebiliyoruz.

242. Satırda dairenin yönü belirten fonksiyon bulunur.

246. Satırdaki fonksiyon ise iki dairenin koordinatlarına göre objelerin birleşip birleşmediği kontrol eder.

253. Satırda level atlandığı anda geçmesi çok hızlı olan bir daire şekli gösterilir.

259. Satırda, baslat ve levelAtla fonksiyonlarında oyunun uygun halde başlayıp başlamadığını kontrol eden kullandığımız fonksiyon. 5 eklememizin sebebi ellipselerin stroke'a sahip olması.

269. Satırda, draw fonksiyonunda dairelerin frame sınırlarını aşıp aşmadığını kontrol ederek aşarsa oyunu sonlandıran fonksiyonu çağıran fonksiyon.

```
class Buton{
  constructor(harf,x,y,size){
this.harf=harf;
this.x=x;
this.y=y;
this.size=size;
  }
show(){
rectMode(CENTER);
textAlign(CENTER);
strokeWeight(1);
noFill();
stroke(renkFont);
rect(this.x,this.y-10,this.size,this.size);
fill(renkFont);
textFont(myFont);
textSize(this.size-20);
text(this.harf, this.x, this.y);
  }
}
class Ok{
  constructor(x,y,size){
  this.x=x;
  this.y=y;
  this.size=size;
    }
show(){
  noFill();
  strokeCap(ROUND);
  strokeWeight(3);
  beginShape(LINES);
  vertex(this.x,this.y);
  vertex(this.x-this.size,this.y+this.size*2);
  vertex(this.x+this.size,this.y+this.size*2);
  vertex(this.x-this.size,this.y+this.size*2);
  vertex(this.x+this.size,this.y+this.size*2);
  endShape(CLOSE);
  }
}
```

Yukarıdaki iki classta giriş ekranında oyuncunun nasıl oynayacağını gösteren ok ve buton şekillerini iki farklı class yaparak oluştrduk.