

WEB PROGRAMLAMA JAVASCRIPT

ÖDEV 02



Furkan Kaya

191216002

ÖĞRETİM GÖREVLİSİ
OĞUZHAN TAŞ

13 Mayıs 2021

Giriş:

Merhaba, bu rapor BIL-202 kodlu Web programlama Javascript dersinde verilmiş ödev 02 için hazırlanmıştır. İyi okumalar dilerim.

Statik:

Web sitenin yapısal kodunu yazmayla işe başladım. HTML kodları ile siteyi statik olarak kodladım. Daha sonrasında javascript ile dinamik hale getirdim.

```
<center>
  <div class="container">
    <div class="main vertical-center">
      <div id="matrix" class="grid" style="padding:2px">

        <div id="0" class="cell"></div>
        <div id="1" class="cell"></div>
        <div id="2" class="cell"></div>
        <div id="3" class="cell"></div>

        <div id="4" class="cell"></div>
        <div id="5" class="cell"></div>
        <div id="6" class="cell"></div>
        <div id="7" class="cell"></div>

        <div id="8" class="cell"></div>
        <div id="9" class="cell"></div>
        <div id="10" class="cell"></div>
        <div id="11" class="cell"></div>

        <div id="12" class="cell"></div>
        <div id="13" class="cell"></div>
        <div id="14" class="cell"></div>
        <div id="15" class="cell"></div>

      </div>
      <div class="puan-bar">
        Puan:
        <span id="puan">
          0
        </span>
      </div>
    </div>
    <div class="button" onClick="startGame()">
      Start Game
    </div>
  </center>
```

Fonksiyonlar:

CreateMatrix():

HTML kodlarıyla statik olarak oluşturduğum matrix yapısını oluşturmak için bir fonksiyon oluşturdum. Önce matrix divinin içini boşaltıp boyutu ve grid yapısını istenilen ölçüye getiriyorum. Ardından yeteri kadar hücreyi for döngüsü ile bu hücrenin içine yerleştiriyorum.

```
function createMatrix(n, isDummy) {
  $("#matrix").empty();
  $("#matrix").css({
    {
      "grid-template-columns": `repeat(${n}, 1fr)`,
      "grid-template-rows": `repeat(${n}, 1fr)`,
      "width": (n * 20) + (n - 1) * 2,
    }
  });
  n = n * n;
  if (isDummy) {
    for (let i = 0; i < n; i++) {
      $("#matrix").append(`<div id="${i}" class="cell"></div>`);
    }
  } else {
    for (let i = 0; i < n; i++) {
      $("#matrix").append(`<div id="${i}" onClick="cellClick(${i})" class="cell"></div>`);
    }
  }
}
```

SelectCell():

CreateMatrix fonksiyonu ile oluşturduğum grid yapısındaki hücrelerden birini rastgele seçip renklendirdiğim fonksiyon. Daha sonrasında rahatça bulabilmek için ayrıca selected sınıfını da seçilen hücreye ekliyorum.

```
function SelectCell() {
  cell = $(".cell").shuffle().slice(0, 1);
  randomColor = '#' + ('000000' + Math.floor(Math.random() * 16777215).toString(16)).slice(-6);
  cell.css({ "background-color": randomColor });
  cell.addClass("selected");
}
```

CellClick(id):

Hücrelere tıklandığında çalışacak olan fonksiyon. Gerekli açıklamalar satırların üzerinde yorum satırı olarak yapılmıştır.

```
function cellClick(id) {  
    //kullanıcı bu tur tıklama yaptıysa çık  
    if (isClickedOne) return;  
    //tıklama yapılmasını işaretle  
    isClickedOne = true;  
    //renkli kareye mi tıklandı?  
    if ($("#" + id).hasClass("selected")) {  
        //puanına +10 ekle  
        updatePoint(10)  
    } else {  
        //puanından -5 çıkar  
        updatePoint(-5)  
    }  
}
```

UpdatePoint(p):

Global puan değişkenine puanı ekle ve puan spanının içine yeni değeri koy.

```
function updatePoint(p) {  
    point += p;  
    $("#puan").html(point);  
}
```

Game(speed, matrixSize):

```
point = 0;
isClickedOne = true;
var timeOutID;
function game(speed, matrixSize) {
    //eğer sıradaki tur geldiyse ve kullanıcı tıklama yapmadıysa
    if (!isClickedOne) updatePoint(-5);

    createMatrix(matrixSize);
    SelectCell();
    //Tıklama hakkını resetle
    isClickedOne = false;

    if (point >= 100) {
        timeOutID = setTimeout(game.bind(null, 800, 6), speed);
    } else {
        timeOutID = setTimeout(game.bind(null, 1000, 4), speed);
    }
}
```

Oyun döngüsünün geçtiği fonksiyon recursive algoritma oluşturup kodladım. Bir önceki tur tıklama yapıldı mı yapılmadı mı kontrol edip yapılmadıysa ceza -5 puan atanıyor. Hücreler oluşturulup arasından biri seçiliyor ve ardından eğer oyuncunun puanı 100'ü geçtiyse aşama 2 de olan 800 ms hızında ve 6 matrix boyutunda bir oyun başlıyor.

StartGame():

```
function startGame() {
    clearTimeout(timeOutID);
    point = 0;
    updatePoint(0);
    isClickedOne = true;

    game(1000, 4);
}
```

Tüm oyun bilgilerini sıfırlayıp recursive fonksiyonu da durdurduktan sonra yeni recursive döngü başlatmak için start game butonuna basınca çalışacak fonksiyon.

Aşama 3:

Aşama 3'ü oluşturmak için aşama 1-2 htmlini kopyalayıp aşağıdaki kodlar eklenmiştir.

Vuruş Sesi:

Html kodları arasına vuruş yapıldığında çalınması için yumruk sesi eklenmiştir.

```
<audio id="punch">  
  <source src="asama3_assets/punch-sound.wav" type="audio/mpeg">  
</audio>
```

Ayrıca CellClick fonksiyonu içine yandaki kodlar eklenmiş jquery ile ses oynatımı yapılmıştır.

```
//ses oynat  
$('audio#punch')[0].currentTime = 0;  
$('audio#punch')[0].play();
```

Fare Fotoğrafı:

Fare fotoğrafını seçilmiş hücreye eklemek için selected sınıfının css kodları şöyle düzenlenmiştir.

```
.selected {  
  background-image: url("asama3_assets/mouse-head.png");  
  background-size: cover;  
}
```

İmleç yerine Balyoz:

<https://game-icons.net/1x1/delapouite/thor-hammer.html> sitesini kullanarak 2 adet svg oluşturulup boyutları 64x64px e düşürülmüştür. ilki üzerinde dururken çıkacak olan svg diğeri de hücreye tıklandığında çıkacak olan svg. Bu sayede vuruyormuş hissi oluşturulmuştur. Cell sınıfının css kodlarına cursor olarak bu iki svg eklenmiştir.

