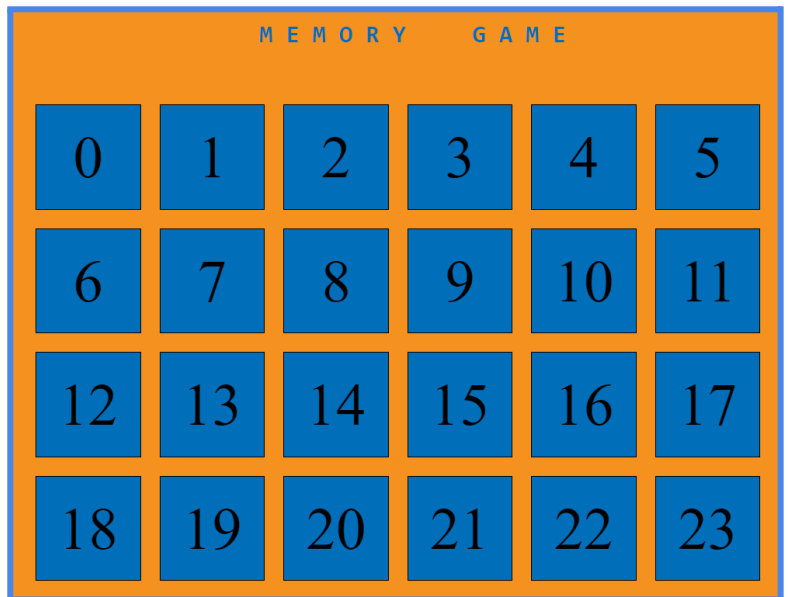

Igra 5

OBJAŠNENJE KAKO RADI IGRICA:

- Igrač otkriva 2 pločice
- Ako su iste, one ostaju otvorene, a ako nisu, one se zatvaraju
- Svaki potez se broji, i svaki uspošno spojen par se broji
- Igricu radimo 2 časa
- Prvi deo igre pamćenja obuhvata prikaz tabele i njenu stilizaciju
- Koriste se sve naredbe koje su do sada naucene i koristene
- Druči deo časa će biti javascript i logika igrice



IDEJA ZA IMPLEMENTACIJU:

- Div u kome ce biti 24 mala diva, ce se dinamicki popunjavati
- Pri kreiranju html dokumenta postojace javascript koji pravi tu tabelu kroz jednu funkciju
- Dodati naslov i polja, labele, u koje ce se setovati vrednosti za broj pokusaja i broj uspjesnih pogadjanja
- Dodati css za glavni div memory board
- Dodati css za male divove unutar memory boarda

POSTUPAK PRAVLJENJA IGRE:

Html fajl:

```
<!DOCTYPE html>
<html style="background-color: #f7931e;">
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="Cas_09_MemoryGame.css" />
  <script src="Cas_09_MemoryGame.js"></script>
</head>
<body>
  <div id="start" style="width: 300px; margin-left: auto; margin-right:
auto;">
```

```

    <h1 style="text-align: center; color: #0071bc;"><pre>M E M O R Y
G A M E</pre></h1>
    <div style="width: 100px; margin-left: auto;margin-right: auto;">
        <label id="rez" style="text-align:center;color:#0071bc;">
        </label>
        <label id="pok" style="text-align:center;color:#0071bc;">
        </label>
        <label id="poeni" style="text-align:center;color:#0071bc;">
        </label>
    </div>
</div>
<div id="memory_board">
</div>
<script>newBoard();</script>
</body>
</html>

```

JS fajl:

```

function newBoard(){
    var output = '';
    for(var i = 0; i < 24; i++){
        output += '<div>'+i+'</div>';
    }
    document.getElementById('memory_board').innerHTML = output;
}

```

CSS fajl:

```

div#memory_board{
    background:#f7931e;
    width:800px;
    height:540px;
    padding:24px;
    margin:0px auto;
}
div#memory_board > div{
    background: #0071bc;
    border:#000 1px solid;
    width:71px;
    height:71px;
    float:left;
}

```

```
margin:10px;
padding:20px;
font-size:64px;
cursor:pointer;
text-align:center;
}
```

OBJASNJENJE ZADATKA:

NAPOMENA 1: dinamičko kreiranje divova - pogledati u inspectu browsera i videti da su svi divovi tu

NAPOMENA 2: probati sa više od 24 iteracije

NAPOMENA 3: prebaciti stilizaciju naslova i labela u css fajl

DEO 2 IMPLEMENTACIJA:

- Dodati globalne promenljive u js fajl - memory array, memory values, memory tile id, tiles flipped, score, attempts
- Dodati funkciju za mešanje tajlova, i pozvati je u newBoard() funkciji
- Dodati funkciju za klik na div na dva mjesta - definisati tu funkciju, i pozvati je u newBoard() funkciji
- Dodati funkciju za okretanje nazad
- Dodati funkciju za azuriranje scorea i pokušaja
- Dodati pozive za tu funkciju u newBoard(), i u okretanju tajlova

POSTUPAK:

Html: Bez promene

CSS: Bez promene

JS: podebljano je sve što je novo u fajlu

- 1) Funkcija za mešanje elemenata niza sa klasičnim swap-om elemenata. Ta funkcija će se pozivati prilikom kreiranja boarda.
- 2) Onclick metoda, dodavanje i definisanje šta će da radi -
- 3) Funkcija za ažuriranje pokušaja i spojenih tajlova, koja će se pozivati u svakom potezu

```
var memory_array =
['A', 'A', 'B', 'B', 'C', 'C', 'D', 'D', 'E', 'E', 'F', 'F', 'G', 'G', 'H', 'H', 'I', 'I',
, 'J', 'J', 'K', 'K', 'L', 'L'];
var memory_values = [];
var memory_tile_ids = [];
var tiles_flipped = 0;
var score = 0
```

```

var attempts = 0

// ovakvo pravljenje funkcija nije radjeno ranije
Array.prototype.memory_tile_shuffle = function(){
    var i = this.length, j, temp;
    while(--i > 0){
        j = Math.floor(Math.random() * (i+1));
        temp = this[j];
        this[j] = this[i];
        this[i] = temp;
    }
}

function newBoard(){
    tiles_flipped = 0;
    score = 0
    attempts = 0
    var output = '';
    memory_array.memory_tile_shuffle();
    for(var i = 0; i < memory_array.length; i++){
        output += '<div id="tile_'+i+'"
onclick="memoryFlipTile(this,\''+memory_array[i]+'\'></div>';
    }
    document.getElementById('memory_board').innerHTML = output;
    updateScore(attempts, score)
}

function updateScore(att, score) {
    content = "Pokušaji: <b>" + att + "</b>";
    document.getElementById('rez').innerHTML = content;
    content = "Spojeni: <b>" + score + "</b>";
    document.getElementById('pok').innerHTML = content
}

function memoryFlipTile(tile, val){
    if(tile.innerHTML == "" && memory_values.length < 2){
        tile.style.background = '#FFF';
        tile.innerHTML = val;
        if(memory_values.length == 0){
            memory_values.push(val);
            memory_tile_ids.push(tile.id);
        } else if(memory_values.length == 1){
            memory_values.push(val);
            memory_tile_ids.push(tile.id);
            attempts +=1;
        }
    }
}

```

```

// ako smo pogodili par
if(memory_values[0] == memory_values[1]){
    tiles_flipped += 2;
    score += 1;
    // resetuj oba niza
    memory_values = [];
    memory_tile_ids = [];
    // spec slucaj, da li je kraj igre
    if(tiles_flipped == memory_array.length){
        alert("Sve je upareno... generisem novi board");
        document.getElementById('memory_board').innerHTML="";
        newBoard();
    }
} else {
    // ako nismo pogodili, okreni ih nazad
    function flip2Back(){
        // Flipujemo nazad tajlove
        var tile_1=document.getElementById(memory_tile_ids[0]);
        var tile_2=document.getElementById(memory_tile_ids[1]);
        tile_1.style.background = "#0071bc";
        tile_1.innerHTML = "";
        tile_2.style.background = "#0071bc";
        tile_2.innerHTML = "";
        // resetujemo oba niza
        memory_values = [];
        memory_tile_ids = [];
    }
    setTimeout(flip2Back, 700);
}
updateScore(attempts, score)
}
}
}

```

OBJASNJENJE ZADATKA:

NAPOMENA 1: Izdvojiti flip back u zasebnu funkciju

NAPOMENA 2: Okretanje nazad je zapravo setovanje vrednosti na "" i backgrounda na plavu boju

NAPOMENA 3: Napraviti dodatno dugme new game da postoji pored boarda da moze da se resetuje igra

NAPOMENA 4: Resetovanje je zapravo da se innerHTML setuje na "" pa onda da se pozove newBoard()

NAPOMENA 5: Uporedjivanje da li se otvorio isti tajl je uporedjivanje memory valuesa

NAPOMENA 6: onclick funkcija dobija 2 argumenta - i objekat na koji je kliknuta (this) i vrednost koju ona nosi (i-ti indeks u promesanoj listi)