



RELATÓRIO DO JOGO PAR OU ÍMPAR

DOCUMENTO ELABORADO POR:

Hugo Lopes

UFCD - 10791

Índice

Introdução	2
Explicação do ficheiro de HTML.....	3
Explicação do ficheiro de JavaScript.....	6
Explicação do ficheiro de CSS.....	12
Conclusão.....	20

Introdução

Este documento insere-se no âmbito da UFCD 10791 – Desenvolvimento de aplicações web em JAVA e tem como intuito descrever o código criado para o projeto final. O documento descreve de forma pormenorizada as linhas de código apresentadas nos ficheiros de HTML, CSS e JavaScript. Os ficheiros em questão representam a criação de um website com o jogo Par ou Ímpar, inserindo-se na continuidade dos projetos realizados nas UFCD anteriores de algoritmos e programação em JAVA.

Explicação do ficheiro de HTML

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Jogo do Par ou Impar</title>
  <link rel="stylesheet" href="style.css">
</head>
```

Definição do título, estilo css, escala e o sistema de codificação para Unicode.

```
<h1>Jogo do Par ou Impar</h1>
<div class="game-container">
  <div class="player-selection">
    <h2>Jogador 1:</h2>
    <img id="player1-image" src="" alt="Selected Image">
    <div class="dropdown">
      <button class="dropbtn">Escolhe a tua personagem</button>
      <div class="dropdown-content">
        <a href="#">
          
        </a>
        <a href="#">
          
        </a>
        <a href="#">
          
        </a>
      </div>
    </div>
  </div>
</div>
```

Escolha da personagem com um dropdown, através da seleção de 3 possíveis imagens.

```

<div class="player-selection">
  <h2>Jogador 2:</h2>
  <img id="player2-image" src="" alt="Selected Image">
  <div class="dropdown">
    <button class="dropbtn">Escolhe a tua personagem</button>
    <div class="dropdown-content">
      <a href="#">
        
      </a>
      <a href="#">
        
      </a>
      <a href="#">
        
      </a>
    </div>
  </div>
</div>

```

Escolha da personagem para o segundo jogador.

```

</div>
<input type="radio" id="player2-human" name="player2" checked>
<label for="player2-human">Jogador</label>
<input type="radio" id="player2-computer" name="player2">
<label for="player2-computer">Computador</label>

```

Possibilidade de jogo contra o “Computador”.

```

<button id="play-button" class="play-button">Joga</button>
<button id="reset-button" class="reset-button">Reinicia</button>

```

Definição do início e fim do jogo.

```
<div class="scoreboard">  
  <h2>Tabela de Resultados</h2>  
  <p id="player1-score">Jogador 1: 0</p>  
  <p id="player2-score">Jogador 2: 0</p>  
</div>
```

Criação de uma tabela de resultados.

```
<script src="script.js"></script>
```

Referência para o ficheiro de JavaScript.

Explicação do ficheiro de JavaScript

```
const gameContainer = document.querySelector('.game-container');

const playButton = document.querySelector('#play-button');
const resetButton = document.querySelector('#reset-button');
const gameMessageElement = document.querySelector('#game-message');

const player1ScoreElement = document.querySelector('#player1-score');
const player2ScoreElement = document.querySelector('#player2-score');

const player1ChoiceElement = document.querySelector('#player1-choice');
const player2ChoiceElement = document.querySelector('#player2-choice');
const player1ChoiceElement: Element | null;

const evenButton1 = player1ChoiceElement.querySelector('.even-button');
const oddButton1 = player1ChoiceElement.querySelector('.odd-button');

const player1ChoiceResultElement = document.querySelector('#player1-choice-result');
const player2ChoiceResultElement = document.querySelector('#player2-choice-result');

const gameResultElement = document.querySelector('#game-result');

const dropdown1 = document.querySelector('.player-selection:nth-child(1) .dropdown');
const dropdown2 = document.querySelector('.player-selection:nth-child(2) .dropdown');

const selectedImage1 = document.getElementById('player1-image');
const selectedImage2 = document.getElementById('player2-image');

const dropdownContent1 = document.querySelector('.player-selection:nth-child(1) .dropdown-content');
const dropdownContent2 = document.querySelector('.player-selection:nth-child(2) .dropdown-content');

const images1 = document.querySelectorAll('.player-selection:nth-child(1) .dropdown-content img');
const images2 = document.querySelectorAll('.player-selection:nth-child(2) .dropdown-content img');

let player1Type = 'human';
let player2Type = 'human';
let currentPlayer = 1;
let gameStarted = false;
let score = { player1: 0, player2: 0 };
let player1Choice;
let player2Choice;
```

Definição das variáveis utilizadas. Como por exemplo a definição de variáveis botões (play, reset, even, odd), seleção das escolhas dos jogadores, imagens utilizadas, entre outros. Definição do tipo de jogador, início do jogo, resultados (score) e escolhas dos jogadores.

```

playButton.addEventListener('click', iniciaJogo);
resetButton.addEventListener('click', reiniciaJogo);
evenButton1.addEventListener('click', () => {
    evenButton1.classList.add('pressed');
    player1Choice = true;
    jogo();
});
oddButton1.addEventListener('click', () => {
    oddButton1.classList.add('pressed');
    player1Choice = false;
    jogo();
});

```

Atribuição de funções aos botões , aguardando que sejam pressionados.

```

function iniciaJogo() {
    gameStarted = true;
    let player2TypeRadio = document.querySelector('input[name="player2"]:checked');
    if (player2TypeRadio.id === 'player2-human')
        player2Type = 'human';
    else
        player2Type = 'computer';
    player1Type = 'human'; // sempre humano
    jogo();
}

```

Função que dá início ao jogo, onde é definido o tipo de jogador. Após a definição é chamada a função jogo. Esta função é chamada quando o botão início é pressionado.


```
function reiniciaJogo() {  
    gameStarted = false;  
    score = { player1: 0, player2: 0 };  
    player1ScoreElement.textContent = 'Player 1: 0';  
    player2ScoreElement.textContent = 'Player 2: 0';  
    gameMessageElement.textContent = '';  
    player1ChoiceResultElement.textContent = '';  
    player2ChoiceResultElement.textContent = '';  
    gameResultElement.textContent = '';  
}
```

Método utilizado para reiniciar o jogo. Elimina os scores dos jogadores e as escolhas do tipo de jogador. Se o jogador um estava a jogar contra o computador pode agora seleccionar jogar contra outro jogador, ou simplesmente faz um reset aos resultados.

```

function jogo() {
  if (!gameStarted) return;

  let player1Number;
  let player2Number;

  // Get the player's choice (even or odd)
  if (player1Type === 'human') {
  }
  else {
    // Gerar um número aleatório
    player1Number = Math.floor(Math.random() * 100);
    player1Choice = player1Number % 2 === 0;
  }

  // Definir a escolha do jogador 2 após a definição do jogador 1
  player2Choice = !player1Choice;

  // Determinar o vencedor da rodada
  let sum = (player1Type === 'human' ? parseInt(prompt("Jogador 1, por favor insira um número")) : player1Number) + (player2Type === 'human' ?
  parseInt(prompt("Jogador 2, por favor insira um número:")) : player2Number);
  let isEven = sum % 2 === 0;

  // Mostrar as escolhas do jogador
  if (player1Choice) {
    player1ChoiceResultElement.textContent = 'Jogador 1 escolheu par';
    player1ChoiceResultElement.innerHTML += '';
  }
  else {
    player1ChoiceResultElement.textContent = 'Jogador 1 escolheu ímpar';
    player1ChoiceResultElement.innerHTML += '';
  }

  if (player2Choice) {
    player2ChoiceResultElement.textContent = 'Jogador 2 escolheu par';
    player2ChoiceResultElement.innerHTML += '';
  }
  else {
    player2ChoiceResultElement.textContent = 'Jogador 2 escolheu ímpar';
    player2ChoiceResultElement.innerHTML += '';
  }

  // Atualizar a tabela de resultados
  if ((player1Choice && isEven) || (!player1Choice && !isEven)) {
    player1ScoreElement.textContent = `Jogador 1: ${parseInt(player1ScoreElement.textContent.split(":")[1]) + 1}`;
    console.log('Jogador 1 wins this round!');
    updateScoreColors();
  }
  else {
    player2ScoreElement.textContent = `Jogador 2: ${parseInt(player2ScoreElement.textContent.split(":")[1]) + 1}`;
    console.log('Jogador 2 wins this round!');
    updateScoreColors();
  }

  function updateScoreColors() {
    const player1Score = parseInt(player1ScoreElement.textContent.split(":")[1]);
    const player2Score = parseInt(player2ScoreElement.textContent.split(":")[1]);

    if (player1Score > player2Score) {
      player1ScoreElement.style.color = 'green';
      player2ScoreElement.style.color = 'red';
    } else if (player2Score > player1Score) {
      player1ScoreElement.style.color = 'red';
      player2ScoreElement.style.color = 'green';
    } else {
      player1ScoreElement.style.color = 'blue';
      player2ScoreElement.style.color = 'blue';
    }
  }

  resetRound();
  // Jogar a próxima ronda
  setTimeout(() => {
    console.log('Playing next round...');
    player1ChoiceResultElement.textContent = '';
    player2ChoiceResultElement.textContent = '';
    gameResultElement.textContent = '';
  }, 8000);
}

```

Função que faz o controlo de todo o jogo. A primeira linha de código diz-nos que se o jogo não tiver sido iniciado deve logo “saltar fora” e não corre mais nada.

Entre a linha 85 e 90 realizamos a recolha da escolha do jogador, e se este for computador a escolha será fornecida através da criação de um número aleatório.

A linha 92 define que se o jogador um for par o dois é ímpar e vice versa.

Entre a linha 96 e a linha 98 fazemos a avaliação do vencedor através da utilização de um operador ternário (também conhecido como if ternário).

Entre a linha 101 e 117 criamos um novo “menu” temporário que irá mostrar a opção que cada jogador teve, juntamente com uma imagem representando o par ou Ímpar.

Entre a linha 120 e a linha 129 vamos atualizar a tabela de resultados mediante o resultado obtido na última ronda.

Entre a linha 131 e 135 criamos uma função que define as cores das letras dos scores dos jogadores. Azul em caso de empate, verde para quem esteja a vencer e vermelho para quem estiver a perder.

Entre a linha 147 e a linha 153 definimos um temporizador que vai mostrar os resultados obtidos nesta ronda durante 8 segundos e de seguida irá ocultar.

No fim chamamos a função resetRound.

```
function resetRound() {  
    evenButton1.classList.remove('pressed');  
    oddButton1.classList.remove('pressed');  
}
```

Função que faz reinício da ronda, basicamente removendo o press do botão par ou do botão ímpar.

```
dropdown1.addEventListener('click', () => {
  dropdownContent1.classList.toggle('show');
});

dropdown2.addEventListener('click', () => {
  dropdownContent2.classList.toggle('show');
});
```

Dropmenu para a escolha das personagens iniciais. (Ficou no fim do código pois foi uma ideia que surgiu após ter todo o programa a funcionar.

```
images1.forEach((image) => {
  image.addEventListener('click', () => {
    selectedImage1.src = image.src;
    dropdownContent1.classList.remove('show');
  });
});

images2.forEach((image) => {
  image.addEventListener('click', () => {
    selectedImage2.src = image.src;
    dropdownContent2.classList.remove('show');
  });
});
```

Definição da imagem selecionada após a escolha do utilizador.

Explicação do ficheiro de CSS

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    background-color: #fedf95;  
}
```

Formatação geral do corpo do HTML.

```
.game-container {  
    max-width: 1400px;  
    margin: 40px auto;  
    padding: 10px;  
    border: 1px solid #ccc;  
    border-radius: 10px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    background-image: url('ParOuImpar.png');  
    background-size: cover;  
    background-position: center;  
    background-attachment: fixed;  
}
```

Formatação do container do jogo onde apresenta uma imagem de fundo central.

```
.player-selection {  
  margin-bottom: 20px;  
}  
  
.game-board {  
  margin-bottom: 20px;  
}  
  
.scoreboard {  
  margin-top: 20px;  
}
```

Definição das margens para a tabela de resultados, a tabela de jogo e seleção do jogador.

```

button {
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  background-color: #fedf95;
  color: #000000;
  cursor: pointer;
  font-size: 16px;
}

button:hover {
  background-color: #d5c492;
}

.play-button, .reset-button {
  font-size: 18px;
}

.play-button:hover {
  background-color: #3e8e41;
}

.reset-button {
  background-color: #e74c3c; /* Vermelho */
  color: #000000;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.reset-button:hover {
  background-color: #c0392b;
}

```

Definições dos botões: Cores, tamanhos, margens, entre outros.

```
.choice-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
  margin-top: 20px;
}
```

Formatação do container da escolha.

```
.even-button {
  background-color: #ae0898;
  color: #000000;
}

.odd-button {
  background-color: #ff69b4;
  color: #000000;
}

.pressed {
  background-color: #ccc;
  color: #666;
  transform: translateY(2px);
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

.even-button:hover {
  background-color: #4e96a4;
}

.odd-button:hover {
  background-color: #ebc7d9;
}
```

Definição dos botões de par ou ímpar com definição para quando o rato passar por cima e para quando o botão se encontra selecionado.


```
.result-container {  
  margin-top: 20px;  
  font-size: 18px;  
}
```

Formatação do container dos resultados.

```
#player1-choice-result {  
  font-size: 18px;  
  font-weight: bold;  
  margin-bottom: 10px;  
}  
  
#game-result {  
  font-size: 24px;  
  font-weight: bold;  
  color: #fedf95;  
  margin-bottom: 20px;  
}
```

Formatação da escolha do jogador e dos resultados do jogo

```
.even-image {  
  width: 20px;  
  height: 20px;  
  margin: 5px;  
}  
  
.odd-image {  
  width: 20px;  
  height: 20px;  
  margin: 5px;  
}
```

Definição do tamanho das imagens do par ou ímpar.

```

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content img {
  width: 50px;
  height: 50px;
  margin: 5px;
}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {
  display: block;
}

.dropdown:hover .dropbtn {
  background-color: #3e8e41;
}

```

Formação do Dropdown menu: Tamanhos cores, posição, entre outros.

```
#player1-image, #player2-image {  
  width: 50px;  
  height: 50px;  
  border-radius: 50%;  
}
```

Formatação das imagens da personagem escolhida pelo jogador.

```
#player1-score, #player2-score {  
  font-size: 24px;  
  font-weight: bold;  
}
```

Formatação da tabela de resultados.

```
@media (max-width: 900px) {  
  .game-container {  
    max-width: 800px;  
  }  
  .choice-container {  
    width: 70%;  
  }  
  .choice {  
    width: 35%;  
  }  
}  
  
@media (max-width: 600px) {  
  .game-container {  
    max-width: 600px;  
  }  
  .choice-container {  
    width: 80%;  
  }  
  .choice {  
    width: 40%;  
  }  
}  
  
@media (max-width: 400px) {  
  .game-container {  
    max-width: 400px;  
  }  
  .choice-container {  
    width: 90%;  
  }  
  .choice {  
    width: 45%;  
  }  
}
```

Definições responsivas do website.

Conclusão

A realização deste projeto demonstrou continuidade evolutiva no processo de aprendizagem de Java. Consegui colmatar algumas falhas dos projetos das UFCD anteriores e conseguir acrescentar alguma parte gráfica, que torna o jogo mais apelativo. Com a conclusão do mesmo pude confirmar a minha progressão na ação de formação.