

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METRÓPOLE DIGITAL – IMD

Disciplina: Desenvolvimento de Sistemas Web I

Docente: André Luiz Da Silva Solino

Discentes: Pedro Lucas Gois Costa, Renato Galvão da Silva, Henrique Eduardo Martins da Silva Filho

Projeto da unidade 3 - Jogo da Velha

O projeto final da unidade 3 consiste no desenvolvimento de um jogo da velha no formato de aplicação web, utilizando as tecnologias e linguagens aprendidas em sala de aula (HTML, CSS e Javascript). Este relatório visa descrever o processo de desenvolvimento do projeto de forma a explicar as funcionalidades, formatação e estilização detalhadamente.

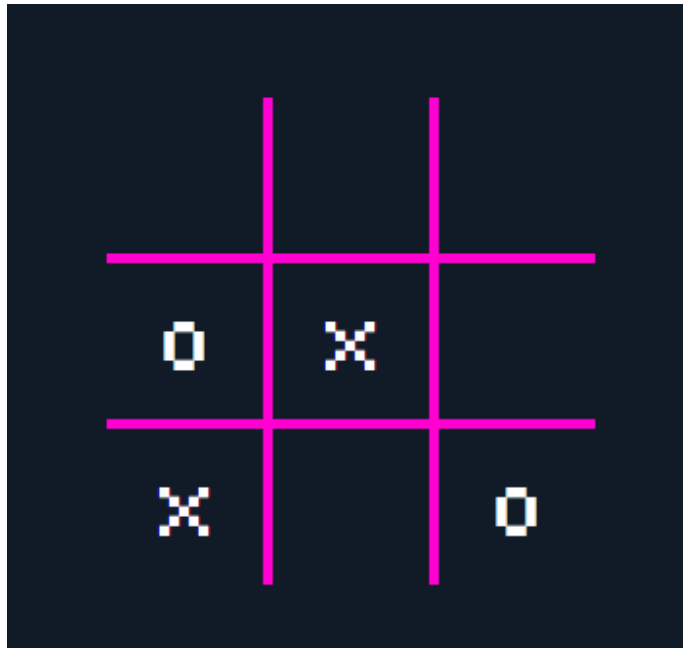
1. Funcionalidades implementadas

Todas as funcionalidades solicitadas no documento foram implementadas, e abaixo estão descritas uma a uma.

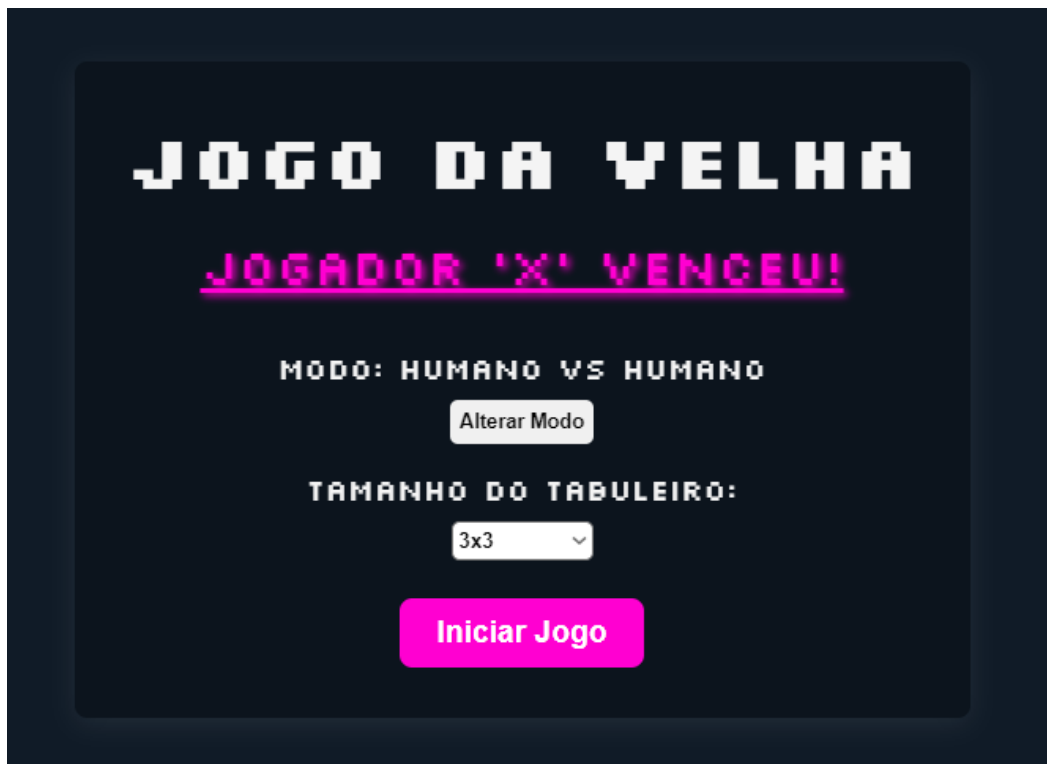
1.1 - Menu inicial, contendo as configurações da partida. Aqui o usuário poderá selecionar se irá jogar contra outro jogador ou contra o computador (algoritmo desenvolvido para simular um robô), como também escolher o tamanho do tabuleiro que varia de 3x3 até 10x10. Para inicializar o jogo basta selecionar as configurações desejadas e apertar no botão “Iniciar Jogo”



1.2 - Tabuleiro da partida onde o jogo acontece, nesta tela é possível selecionar a cada turno do jogador uma célula do tabuleiro, que será marcada com o símbolo do jogador da vez (X ou O).



1.3 - No momento que um jogador vence a partida (Selecione células do tabuleiro de forma que elas formem uma linha reta na vertical, horizontal ou até mesmo na diagonal), o menu volta a aparecer com a mensagem do jogador vencedor. Caso nenhum dos jogadores vença, uma mensagem de empate é mostrada na tela.



2. Estrutura do código

O código do projeto foi modularizado em 3 partes, um arquivo para cada linguagem (index.html, style.css e script.js).

2.1 - A estrutura principal da aplicação está contida no arquivo HTML. Neste são inseridos os trechos de:

- **Cabeçalho**, contendo desde o título da página, a importação da fonte utilizada, como também a importação do arquivo css que está separado.
- **Corpo da página**, contendo a estrutura do menu do jogo e também a <div> que servirá de container para o jogo ser criado através do javascript.
- **Script**, que neste caso serve para importar o javascript que está separado.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Silkscreen:wght@400;700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="style.css">
    <title>Jogo da Velha</title>
  </head>
  <body>
    <main>
      <div id="game-info">
        <div>
          <h1>Jogo da velha</h1>
          <p id="result"></p>
          <label id="mode-info">Modo: Humano vs Humano</label>
          <button id="mode-toggle">Alterar Modo</button>
          <label for="board-size">Tamanho do Tabuleiro:</label>
          <select id="board-size">
            <option value="3">3x3</option>
            <option value="4">4x4</option>
            <option value="5">5x5</option>
            <option value="6">6x6</option>
            <option value="7">7x7</option>
            <option value="8">8x8</option>
            <option value="9">9x9</option>
            <option value="10">10x10</option>
          </select>
          <button id="start-button">Iniciar Jogo</button>
        </div>
      </div>
      <div class="game-container">
        <div id="game-board"></div>
      </div>
    </main>
    <script src="script.js"></script>
  </body>
</html>
```

2.2 - A estilização da aplicação está contida no arquivo CSS. Nele foram utilizadas marcações de classes e ID's para atribuir propriedades de estilo para cada elemento html do site.

2.3 - Os scripts que fazem a aplicação funcional estão contidos no arquivo JS. Nele foram criados diversas funções com o intuito de fazer o jogo funcional e interativo.

- **startGame()** - Esta função tem como principal intuito limpar e criar as células do tabuleiro para que um novo jogo se inicie

```
function startGame() {
  currentPlayer = 'X';
  gameBoard = Array.from({ length: boardSize * boardSize }, () => '');
  gameActive = true;

  // Limpar o tabuleiro
  boardContainer.innerHTML = '';

  // Atualizar o estilo do tabuleiro
  boardContainer.style.gridTemplateColumns = `repeat(${boardSize}, 80px)`;

  // Criar células do tabuleiro dinamicamente
  for (let i = 0; i < boardSize * boardSize; i++) {
    const cell = document.createElement('div');
    cell.classList.add('cell');
    cell.dataset.index = i;
    cell.addEventListener('click', () => handleCellClick(i));
    boardContainer.appendChild(cell);
  }

  // Reiniciar o jogo
  if (againstComputer && currentPlayer === 'O') {
    setTimeout(makeComputerMove, 500);
  }

  gameInfo.classList.add('hidden')
}
```

- **makeMove()** - Esta armazena o local onde o jogador/computador selecionou e atualiza o texto da célula selecionada com o símbolo do jogador que a selecionou.

```
function makeMove(index) {
  gameBoard[index] = currentPlayer;
  boardContainer.querySelector(`[data-index="${index}"]`).textContent = currentPlayer;;
}
```

- **handleCellClick()** e **makeComputerMove()** - Ambas as funções servem para realizar a ação de selecionar uma celular quando o jogador clica, entretanto o **makeComputerMove** só é chamada caso o modo de jogo seja Humano x Computador

```
function handleCellClick(index) {
  if (gameActive && !gameBoard[index]) {
    makeMove(index);
    checkWinner();
    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
    if (gameActive && againstComputer && currentPlayer === 'O') {
      setTimeout(makeComputerMove, 500);
    }
  }
}

function makeComputerMove() {
  const availableCells = gameBoard.reduce((acc, val, index) => {
    if (!val) acc.push(index);
    return acc;
  }, []);

  if (availableCells.length > 0) {
    const randomIndex = Math.floor(Math.random() * availableCells.length);
    makeMove(availableCells[randomIndex]);
    checkWinner();
    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
  }
}
```

- **checkWinner()** - Verifica toda rodada se algum jogador venceu. Ele cria todos os padrões possíveis de ganhar o jogo e verifica se algum dos jogadores atingiu esse requisito e por fim retorna a tela de menu com o nome do jogador que venceu ou caso nenhum dos dois vença, com uma mensagem de empate.

```
function checkWinner() {
  const winPatterns = [];
  for (let i = 0; i < boardSize; i++) {
    winPatterns.push(Array.from({ length: boardSize }, (_, j) => i * boardSize + j));
    winPatterns.push(Array.from({ length: boardSize }, (_, j) => i + j * boardSize));
  }
  winPatterns.push(Array.from({ length: boardSize }, (_, i) => i * (boardSize + 1)));
  winPatterns.push(Array.from({ length: boardSize }, (_, i) => (i + 1) * (boardSize - 1)));

  for (const pattern of winPatterns) {
    const winner = pattern.every(index => gameBoard[index] === currentPlayer);
    if (winner) {
      gameActive = false;
      result.textContent = `Jogador '${currentPlayer}' venceu!`;
      gameInfo.classList.remove('hidden');
      return;
    }
  }

  if (!gameBoard.includes('')) {
    gameActive = false;
    result.textContent = 'Empate!';
    gameInfo.classList.remove('hidden');
  }
}
```

3. O que faltou ser feito e o que poderia ter sido feito melhor?

No geral todas as implementações solicitadas foram realizadas, entretanto o modo Humano x Computador não contém um algoritmo muito avançado que dificulte o jogo para o Humano. Portanto, algo que poderia ter sido feito melhor é algum algoritmo que faça com que o computador tenha mais chances de ganhar.

4. Como acessar o projeto

O projeto pode ser acessado entrando na pasta do projeto e clicando duas vezes no arquivo index.html. ou então através do link <https://pedrolgois.github.io/jogo-da-velha/>

5. Contribuição dos integrantes

O projeto foi desenvolvido em chamada no discord em conjunto dos 3 participantes. Pedro foi responsável por grande parte da estilização do projeto e os 3 em conjunto pela parte de estrutura HTML e código de Javascript. O relatório foi feito também em conjunto pelos 3