JavaScript Project: Build a Simple Tic-Tac-Toe Game

Session Overview

In this session, students will build a simple Tic-Tac-Toe game using JavaScript. They will apply previous knowledge of DOM manipulation, arrays, loops, conditionals, and event handling.

Learning Objectives

By the end of this session, students will be able to:

- - Use JavaScript to create HTML elements dynamically
- - Apply event listeners for interaction
- - Control game state using logic and arrays
- - Reset and restart a DOM-based game

Materials Needed

- A simple HTML file with a board container

- A linked JS file for the game logic

Step 1: Basic HTML Structure

```
<!DOCTYPE html>
<html>
<head>
  <title>Tic-Tac-Toe</title>
  <style>
    .board { display: grid; grid-template-columns: repeat(3, 100px); gap: 5px; }
    .cell { width: 100px; height: 100px; font-size: 2em; display: flex; justify-content: center;
align-items: center; border: 1px solid black; cursor: pointer; }
  </style>
</head>
<body>
  <h1>Tic-Tac-Toe Game</h1>
  <div id="board" class="board"></div>
  <p id="status"></p>
```

```
  <button onclick="resetGame()">Reset</button>
  <script src="script.js"></script>
</body>
</html>
```

💻 Step 2: JavaScript Code

```
let board = ['', '', '', '', '', '', '', '', ''];
let currentPlayer = 'X';
let isGameActive = true;
const boardElement = document.getElementById('board');
const statusElement = document.getElementById('status');

function drawBoard() {
  boardElement.innerHTML = '';
  board.forEach((cell, index) => {
    const cellDiv = document.createElement('div');
    cellDiv.classList.add('cell');
    cellDiv.textContent = cell;
    cellDiv.addEventListener('click', () => makeMove(index));
    boardElement.appendChild(cellDiv);
  });
}

function makeMove(index) {
  if (board[index] === '' && isGameActive) {
    board[index] = currentPlayer;
    checkWinner();
    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
    drawBoard();
  }
}

function checkWinner() {
  const winConditions = [
    [0,1,2], [3,4,5], [6,7,8],
    [0,3,6], [1,4,7], [2,5,8],
    [0,4,8], [2,4,6]
  ];
  for (const condition of winConditions) {
    const [a, b, c] = condition;
```

```
    if (board[a] && board[a] === board[b] && board[a] === board[c]) {
      statusElement.textContent = `Loja mbaroi! Fituesi: ${board[a]}`;
      isGameActive = false;
      return;
    }
  }
  if (!board.includes('')) {
    statusElement.textContent = 'Barazim!';
    isGameActive = false;
  }
}

function resetGame() {
  board = ['', '', '', '', '', '', '', '', ''];
  currentPlayer = 'X';
  isGameActive = true;
  statusElement.textContent = '';
  drawBoard();
}

drawBoard();
```

Student Tasks

- Test and debug the game

- Customize the game design

- Add additional features like score tracking