Guia 1

Site do jogo da velha
em PHP

Caio Marcio Ribeiro Moreira Souza
Rebeca Meira da Silva

(Caiomarcioribeiroo@gmail.com)
(rebeca.meira98@gmail.com)

RELATÓRIO DE LTP

3D colegios - Rio de janeiro - Nova iguaçu
2024

Índice

Introdução

Este documento é para falar sobre como foi feito o site do jogo da velha em php, html e css. Falaremos de uma maneira simplificada para que entenda sobre o assunto, focando um pouco mais no php do nosso trabalho.

Site

Nosso trabalho consiste na criação de um jogo simples, o clássico jogo da velha, que é bastante popular e fácil de entender. As regras do jogo são bem simples e não exigem muito tempo de aprendizado. No jogo, temos dois jogadores: o jogador um, que usa o símbolo "X", e o jogador dois, que utiliza o símbolo "O". Esses dois jogadores competem entre si para ver quem consegue formar uma linha com três símbolos iguais.

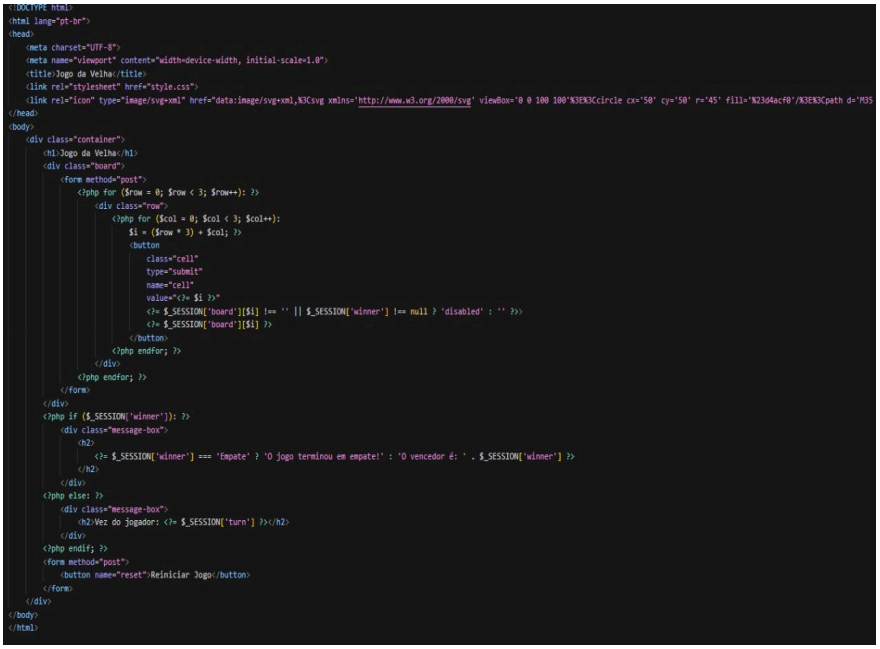
O jogo acontece em um tabuleiro 3x3, que possui nove espaços disponíveis para os jogadores colocarem seus símbolos. O objetivo de cada jogador é preencher uma linha com três de seus símbolos, podendo ser na horizontal, na vertical ou na diagonal. A primeira pessoa que conseguir alinhar seus símbolos vence o jogo. Caso todos os espaços sejam preenchidos e nenhum dos jogadores consiga formar uma linha de três símbolos iguais, o jogo termina em empate. Esse tipo de jogo é ideal para testar raciocínio lógico e tomada de decisões rápidas, além de ser uma excelente forma de diversão para duas pessoas.

HTML

Vamos começar falando da primeira linguagem, que é o HTML. O HTML serve para estruturar o conteúdo de uma página na web. Ele define a estrutura base de um site, organizando textos, fotos, tabelas entre outros. No nosso projeto não foi diferente, montando a base de nosso site.

O código faz o jogo da velha funcionar direto na página, sem precisar ficar recarregando. O PHP cuida de tudo nos bastidores, como manter o controle do tabuleiro, o turno dos jogadores e quem venceu. Ele usa algo chamado "variáveis de sessão", que basicamente permite que o jogo continue de onde parou, sem perder nenhuma informação entre os cliques dos jogadores. Ou seja, quando você clica em um quadrado para jogar, a página é atualizada automaticamente, e as mudanças aparecem ali na hora, sem a necessidade de dar F5.

Já o HTML faz a parte visual, criando o tabuleiro de 3x3, onde você faz suas jogadas, e também mostra as mensagens, tipo "Vez do jogador X" ou "O jogo terminou em empate". Quando um jogador faz uma jogada, o quadrado que ele clicou é desativado, então ninguém consegue jogar ali de novo. No final, o PHP verifica quem venceu ou se deu empate e exibe a mensagem. Tudo isso funciona de forma bem simples e interativa, e ainda tem um botão para reiniciar o jogo, caso alguém queira começar de novo.



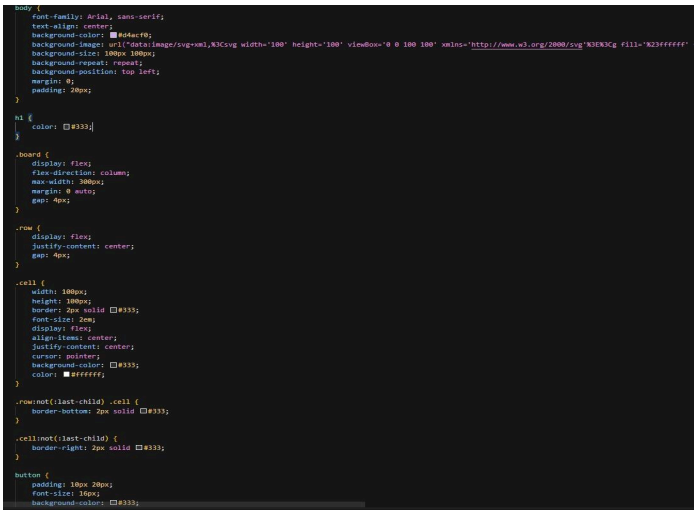
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Jogo da Velha</title>
    <link rel="stylesheet" href="style.css">
    <link rel="icon" type="image/svg+xml" href="data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 100 100'%3E%3Ccircle cx='50' cy='50' r='45' fill='%23d4acf0'/%3E%3Cpath d='M35
</head>
<body>
    <div class="container">
        <h1>Jogo da Velha</h1>
        <div class="board">
            <form method="post">
                <?php for ($row = 0; $row < 3; $row++): ?>
                    <div class="row">
                        <?php for ($col = 0; $col < 3; $col++):
                            $i = ($row * 3) + $col; ?>
                            <button
                                class="cell"
                                type="submit"
                                name="cell"
                                value="<?= $i ?>"
                                <?= $_SESSION['board'][$i] !== '' || $_SESSION['winner'] !== null ? 'disabled' : '' ?>>
                                <?= $_SESSION['board'][$i] ?>
                            </button>
                        <?php endfor; ?>
                    </div>
                <?php endfor; ?>
            </form>
        </div>
        <?php if ($_SESSION['winner']): ?>
            <div class="message-box">
                <h2>
                    <?= $_SESSION['winner'] === 'Empate' ? 'O jogo terminou em empate!' : 'O vencedor é: ' . $_SESSION['winner'] ?>
                </h2>
            </div>
        <?php else: ?>
            <div class="message-box">
                <h2>Vez do jogador: <?= $_SESSION['turn'] ?></h2>
            </div>
        <?php endif; ?>
        <form method="post">
            <button name="reset">Reiniciar Jogo</button>
        </form>
    </div>
</body>
</html>
```

CSS

Esse código de CSS serve para deixar o visual do jogo da velha bem organizado e agradável. O fundo da página tem uma cor lilás suave, com um padrão de imagem que dá um toque especial. A fonte é bem simples e fácil de ler, e o conteúdo é centralizado na tela para não ficar grudado nas bordas. As células do tabuleiro, onde os jogadores clicam, são quadradas e com borda escura para destacar, e quando o jogador passa o cursor, ele vira uma mãozinha, mostrando que a célula é interativa. Além disso, o botão para reiniciar o jogo tem um visual simples, mas elegante, e muda de cor quando o mouse passa por cima, dando um toque de interatividade.

A parte de dentro do jogo, como a caixa onde o tabuleiro fica e as mensagens de vitória ou empate, também é bem estilizada. A caixa do jogo tem fundo branco, bordas arredondadas e uma sombra que faz ela parecer um pouco elevada, o que dá um efeito bacana na tela. As mensagens de quem venceu ou se o jogo deu empate aparecem em uma caixa roxa, com texto branco, o que torna tudo bem visível e fácil de entender. No geral, o código de CSS ajuda a criar uma experiência visual limpa e simples, sem perder a funcionalidade, deixando o jogo fácil de jogar e bonito ao mesmo tempo.

Cores que utilizamos no site: `#8a2594, #ffffff, #333, d4acf0`



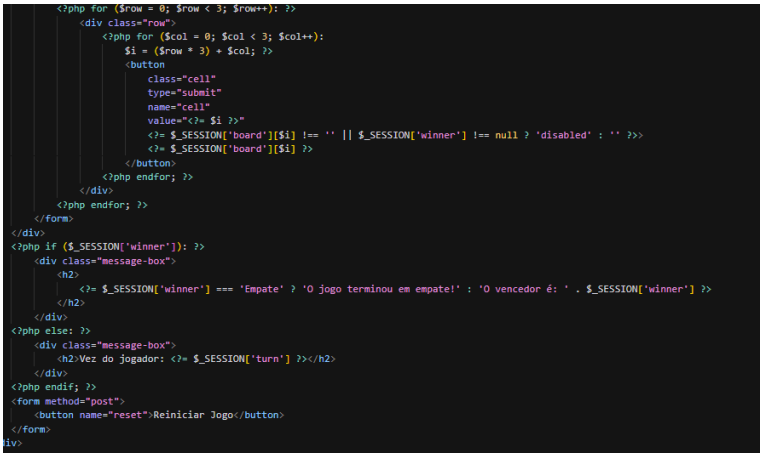
PHP

Agora saindo um pouco da estrutura do site, entramos no funcionamento do site. O PHP podemos dizer que deu a vida ao site, o PHP ele serve para a dinâmica e interação do site, que no nosso caso seria fazer o jogo da velha funcionar.

O código PHP faz o jogo da velha funcionar de forma interativa em uma única página, mantendo tudo em funcionamento sem precisar recarregar a página. Ele usa variáveis de sessão para armazenar as informações do jogo, como o tabuleiro, quem é o próximo jogador e quem venceu, permitindo que o jogo continue sem perder dados entre as jogadas. A cada vez que o jogador faz uma jogada, o PHP atualiza o estado do jogo e atualiza automaticamente a interface HTML, permitindo que o jogador veja o progresso em tempo real, sem a necessidade de atualizar a página manualmente.

Além de controlar o turno dos jogadores, o PHP também cuida das regras do jogo, verificando se alguém ganhou ou se houve empate a cada jogada. Se o jogo terminar, ele mostra a mensagem correspondente (vencedor ou empate) e ainda oferece a opção de reiniciar o jogo. Juntando PHP e HTML, o código cria uma experiência simples e divertida para o usuário, com tudo acontecendo de forma fluída e sem grandes complicações.

```php
<?php
session_start();


if (!isset($_SESSION['board'])) {
    $_SESSION['board'] = array_fill(0, 9, '');
    $_SESSION['turn'] = 'X';
    $_SESSION['winner'] = null;
}

// reseta o jogo
if (isset($_POST['reset'])) {
    session_destroy();
    header("Location: index.php");
    exit();
}
  Ctrl+I to chat, Ctrl+K to generate
// jogada
if (isset($_POST['cell']) && $_SESSION['winner'] === null) {
    $cell = $_POST['cell'];
    if ($_SESSION['board'][$cell] === '') {
        $_SESSION['board'][$cell] = $_SESSION['turn'];
        $_SESSION['turn'] = ($_SESSION['turn'] === 'X') ? 'O' : 'X';
    }
}

// Verifica o
$winning_combinations = [
    [0, 1, 2], [3, 4, 5], [6, 7, 8], // Linhas
    [0, 3, 6], [1, 4, 7], [2, 5, 8], // Colunas
    [0, 4, 8], [2, 4, 6]             // Diagonais
];

// vencedor
foreach ($winning_combinations as $combo) {
    if ($_SESSION['board'][$combo[0]] &&
        $_SESSION['board'][$combo[0]] === $_SESSION['board'][$combo[1]] &&
        $_SESSION['board'][$combo[1]] === $_SESSION['board'][$combo[2]]) {
        $_SESSION['winner'] = $_SESSION['board'][$combo[0]];
    }
}

// empate
if (!in_array('', $_SESSION['board']) && $_SESSION['winner'] === null) {
    $_SESSION['winner'] = 'Empate';
}
?>
```



```
        <?php for ($row = 0; $row < 3; $row++): ?>
            <div class="row">
                <?php for ($col = 0; $col < 3; $col++):
                    $i = ($row * 3) + $col; ?>
                    <button
                        class="cell"
                        type="submit"
                        name="cell"
                        value="<?= $i ?>"
                        <?= $_SESSION['board'][$i] !== '' || $_SESSION['winner'] !== null ? 'disabled' : '' ?>>
                        <?= $_SESSION['board'][$i] ?>
                    </button>
                <?php endfor; ?>
            </div>
        <?php endfor; ?>
    </form>
</div>
<?php if ($_SESSION['winner']): ?>
    <div class="message-box">
        <h2>
            <?= $_SESSION['winner'] === 'Empate' ? 'O jogo terminou em empate!' : 'O vencedor é: ' . $_SESSION['winner'] ?>
        </h2>
    </div>
<?php else: ?>
    <div class="message-box">
        <h2>Vez do jogador: <?= $_SESSION['turn'] ?></h2>
    </div>
<?php endif; ?>
<form method="post">
    <button name="reset">Reiniciar Jogo</button>
</form>
iv>
```

ALGORITMO

Esse algoritmo é a parte central de um jogo da velha simples, mas que funciona direitinho. Ele começa criando um tabuleiro vazio, ou seja, um espaço 3x3 onde os jogadores podem colocar seus símbolos, sendo o 'X' o jogador inicial. O primeiro passo é sempre definir quem vai jogar e, a partir daí, a brincadeira começa. Cada vez que o jogador escolhe uma posição, o algoritmo dá uma olhada para ver se aquele espaço está livre. Se estiver vazio, o símbolo do jogador é colocado no lugar certo no tabuleiro.

Depois de cada jogada, o algoritmo verifica se alguém conseguiu ganhar. Para isso, ele checa todas as possibilidades de vitória, ou seja, as linhas, colunas e as duas diagonais. Se algum jogador tiver preenchido uma dessas com três símbolos iguais, o jogo para imediatamente e o vencedor é anunciado. Caso contrário, o algoritmo alterna o turno, passando a vez para o outro jogador, que será o 'O'. A cada rodada, ele vai atualizando o tabuleiro, e vai verificando novamente as condições de vitória. Isso continua até que um dos jogadores ganhe ou até que o tabuleiro esteja cheio. Se o tabuleiro for preenchido sem que ninguém tenha vencido, o jogo termina em empate. Esse processo é o que mantém o jogo funcionando de forma fluida e garantindo que tudo ocorra de forma justa e divertida.

```
// Jogador escolhe a linha e a coluna
escreva("\nJogador ", jogadorAtual, ", escolha a linha (0 a 2): ")
leia(linha)
escreva("Escolha a coluna (0 a 2): ")
leia(coluna)

// Verifica se a jogada é válida
se tabuleiro[linha][coluna] = ' ' entao
   tabuleiro[linha][coluna] <- jogadorAtual
senao
   escreva("\nPosição ocupada! Tente novamente.")
   continue
fimse

// Verifica se há vencedor
// Linhas
para i de 0 até 2 faca
   se tabuleiro[i][0] = tabuleiro[i][1] e tabuleiro[i][1] = tabuleiro[i][2] e tabuleiro[i][0] != ' ' entao
      vencedor <- tabuleiro[i][0]
   fimse
fimpara

// Colunas
para j de 0 até 2 faca
   se tabuleiro[0][j] = tabuleiro[1][j] e tabuleiro[1][j] = tabuleiro[2][j] e tabuleiro[0][j] != ' ' entao
      vencedor <- tabuleiro[0][j]
   fimse
fimpara
```

```
// Declarando o tabuleiro e variáveis principais
inteiro tabuleiro[3][3], linha, coluna, i, j
caractere jogadorAtual, vencedor

inicio
   // Inicializando o tabuleiro com espaços vazios
   para i de 0 até 2 faca
      para j de 0 até 2 faca
         tabuleiro[i][j] <- ' '
      fimpara
   fimpara

   // Definindo o jogador inicial
   jogadorAtual <- 'X'
   vencedor <- ' '

   // Loop principal do jogo
   enquanto vencedor = ' ' faca
      // Mostra o tabuleiro atualizado
      limparTela()
      escreva("\nTabuleiro atual:\n")
      para i de 0 até 2 faca
         para j de 0 até 2 faca
```

```
// Diagonais
    se tabuleiro[0][0] = tabuleiro[1][1] e tabuleiro[1][1] = tabuleiro[2][2] e tabuleiro[0][0] != ' ' entao
      vencedor <- tabuleiro[0][0]
    fimse

    se tabuleiro[0][2] = tabuleiro[1][1] e tabuleiro[1][1] = tabuleiro[2][0] e tabuleiro[0][2] != ' ' entao
      vencedor <- tabuleiro[0][2]
    fimse

    // Alterna jogador
    se jogadorAtual = 'X' entao
      jogadorAtual <- 'O'
    senao
      jogadorAtual <- 'X'
    fimse
  fimenquanto

  // Mostra o resultado final
  limparTela()
  escreva("\nTabuleiro final:\n")
  para i de 0 até 2 faca
    para j de 0 até 2 faca
      escreva(tabuleiro[i][j], " ")
    fimpara
    escreva("\n")
  fimpara

  escreva("\nParabéns! O jogador ", vencedor, " venceu!\n")

fimalgoritmo
```

FLUXOGRAMA

O fluxograma mostra como o jogo da velha funciona de um jeito bem simples e direto. Primeiro, ele começa configurando o tabuleiro e decidindo quem vai jogar primeiro. Depois, o tabuleiro é exibido, e o jogador faz sua jogada. O sistema então verifica se a jogada é válida. Se estiver tudo certo, o tabuleiro é atualizado com o símbolo do jogador; caso contrário, o fluxo volta para a etapa onde o jogador escolhe novamente.

A cada jogada, o sistema checa se alguém venceu, olhando as linhas, colunas e diagonais. Se um jogador vencer, o jogo para e mostra o vencedor. Se o jogo acabar sem vencedor, o sistema verifica se o tabuleiro está cheio e se terminou em empate. Caso contrário, o jogador é trocado e o jogo continua. O fluxo segue até que o jogo acabe, seja com vitória ou empate, e termina quando não há mais jogadas a serem feitas. Esse processo garante que o jogo aconteça de maneira fluida e sem complicação.