

Introdução

Descrição do jogo e regras:

O Jogo da Velha é um jogo clássico para dois participantes, cujo objetivo é alinhar três símbolos idênticos (X ou O) em sequência na horizontal, vertical ou diagonal de uma matriz 3x3. Cada jogador, de maneira alternada, escolhe uma posição na matriz para colocar o seu símbolo (X ou O). O jogo se encerra quando um dos jogadores consegue completar uma linha com três símbolos iguais, caracterizando uma vitória, ou quando todas as posições da matriz estão preenchidas, resultando em um empate.

Regras:

1. O tabuleiro é composto por uma matriz 3x3.
2. Os jogadores se alternam para escolher uma posição vazia no tabuleiro, inserindo nela seu símbolo.
3. O primeiro jogador a formar uma linha com três símbolos consecutivos (na horizontal, vertical ou diagonal) é declarado vencedor.
4. Caso todas as posições sejam preenchidas sem que haja um vencedor, o jogo termina em empate.

Resultados

Descrição geral do jogo:

Este código implementa um jogo da velha completo em linguagem C. A implementação utiliza uma matriz 3x3 para representar o tabuleiro do jogo e oferece uma interface simplificada no terminal, permitindo que os jogadores escolham suas posições por meio de coordenadas. As funções principais do programa incluem a exibição do tabuleiro atualizado após cada jogada, a verificação das condições de vitória e empate, a alternância entre os jogadores e a validação das entradas para garantir que as jogadas sejam realizadas apenas em posições válidas e desocupadas. Além disso, o programa é estruturado de forma a proporcionar uma experiência de jogo intuitiva e sem erros, com checagens constantes para entradas inválidas e condições de término de partida. Abaixo, apresenta-se um exemplo de uma partida do jogo, demonstrando o funcionamento completo da lógica e das interações entre os jogadores

Exemplificação do código-fonte:

```
#include <stdio.h>
```

```
// Código-fonte completo do jogo da velha
```

```
// ...
```

```
// Função para verificar vitórias e tabuleiro cheio, alternância de jogadores e impressão do tabuleiro
```

Dificuldades encontradas e soluções implementadas

1. Validação das entradas de jogadas:

- Inicialmente, a leitura das coordenadas permitia que os jogadores realizassem jogadas fora dos limites da matriz (por exemplo, em linhas ou colunas com índices maiores que 2). Para corrigir esse problema, foi implementada uma verificação rigorosa das entradas, assegurando que apenas posições válidas e dentro dos limites da matriz 3x3 fossem aceitas. Essa validação impede jogadas inválidas e garante o bom funcionamento do jogo.

2. Verificação de vitórias e empates:

- A determinação do vencedor exige a verificação de todas as linhas, colunas e diagonais da matriz para identificar uma sequência de três símbolos idênticos consecutivos. Para essa finalidade, foi desenvolvida a função **verificarVitoria**, que realiza essas verificações de maneira eficiente. Além disso, o programa verifica se todas as posições do tabuleiro foram preenchidas e, em caso afirmativo, sem que um vencedor tenha sido encontrado, o jogo é declarado como empate.

3. Visualização do tabuleiro:

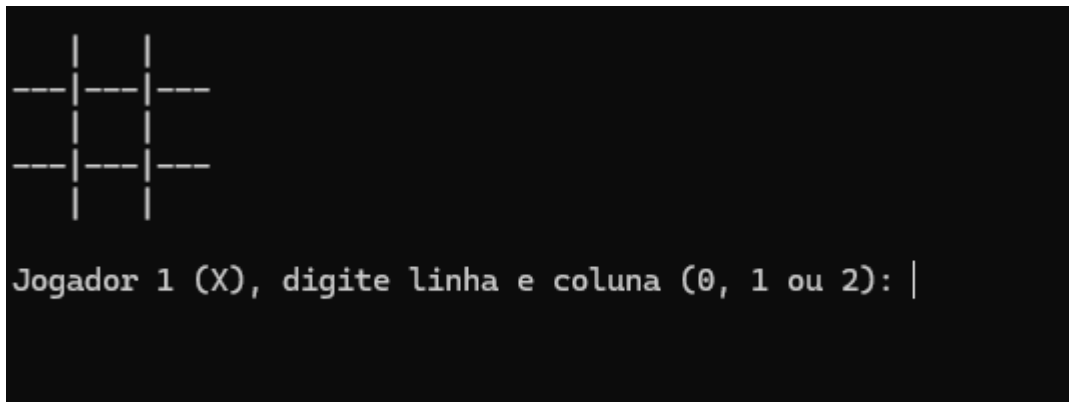
- A fim de aprimorar a experiência visual dos jogadores, foi criada a função **imprimirTabuleiro**, responsável por exibir o estado atual do tabuleiro de maneira clara e organizada. Essa função apresenta divisões nítidas entre as células,

facilitando a leitura e o acompanhamento do progresso do jogo por parte dos participantes.

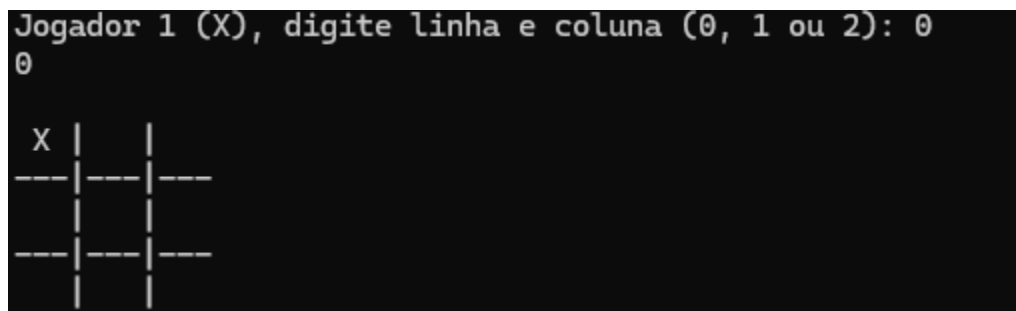
Demonstrativo das funcionalidades implementadas

Aqui estão alguns prints de tela ilustrando o funcionamento do jogo:

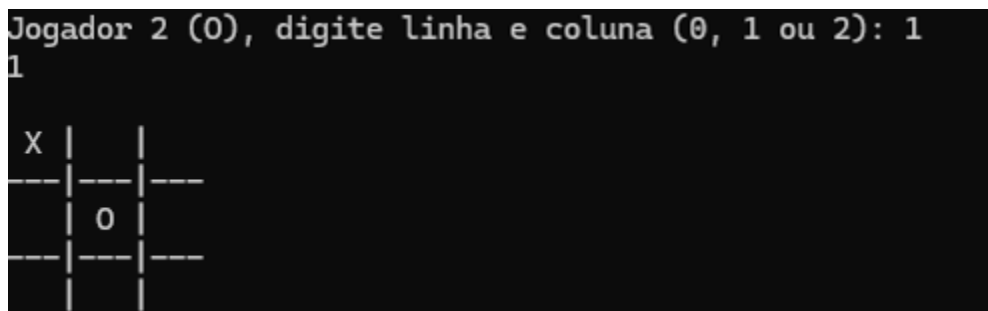
1. Tabuleiro inicial:



2. Jogador 1 (X) faz uma jogada:



3. Jogador 2 (O) faz uma jogada:



4. Final de jogo - Jogador 1 vence:

```
X | X | X
---|---|---
  | O | 
---|---|---
  | O | 
```

Jogador 1 venceu!

Esses prints demonstram a alternância dos jogadores e a atualização do tabuleiro após cada jogada.

Apêndice: Código-fonte completo

Aqui está o código-fonte do jogo da velha em C:

```
#include <stdio.h>
```

```
// Função para imprimir o estado atual do tabuleiro
```

```
void imprimirTabuleiro(char tabuleiro[3][3]) {
```

```
    printf("\n");
```

```
    for (int i = 0; i < 3; i++) {
```

```
        for (int y = 0; y < 3; y++) {
```

```
            printf(" %c ", tabuleiro[i][y]);
```

```
            if (y < 2) printf("|");
```

```
        }
```

```
        printf("\n");
```

```
        if (i < 2) printf("---|---|---\n");
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
// Função para verificar se há um vencedor
```

```
int verificarVitoria(char tabuleiro[3][3]) {
```

```
    for (int i = 0; i < 3; i++) {
```

```
        if (tabuleiro[i][0] == tabuleiro[i][1] &&
```

```
tabuleiro[i][1] == tabuleiro[i][2] && tabuleiro[i][0] != ' ')
```

```
            return 1;
```

```
        if (tabuleiro[0][i] == tabuleiro[1][i] &&
```

```
tabuleiro[1][i] == tabuleiro[2][i] && tabuleiro[0][i] != ' ')
```

```
            return 1;
```

```
    }
```

```
    if (tabuleiro[0][0] == tabuleiro[1][1] &&
```

```
tabuleiro[1][1] == tabuleiro[2][2] && tabuleiro[0][0] != ' ')
```

```
        return 1;
```

```
        if (tabuleiro[0][2] == tabuleiro[1][1] &&  
tabuleiro[1][1] == tabuleiro[2][0] && tabuleiro[0][2] != '  
' )  
            return 1;  
        return 0;  
    }
```

```
// Função para verificar se o tabuleiro está cheio  
int tabuleiroCheio(char tabuleiro[3][3]) {  
    for (int i = 0; i < 3; i++) {  
        for (int y = 0; y < 3; y++) {  
            if (tabuleiro[i][y] == ' ') return 0;  
        }  
    }  
    return 1;  
}
```

```
// Função para limpar o tabuleiro  
void limparTabuleiro(char tabuleiro[3][3]) {  
    for (int i = 0; i < 3; i++) {  
        for (int y = 0; y < 3; y++) {  
            tabuleiro[i][y] = ' ';  
        }  
    }  
}
```

```
// Função principal  
int main() {  
    char tabuleiro[3][3];  
    int linha, coluna, jogador = 1;  
    char simbolo;
```

```
limparTabuleiro(tabuleiro);

while (1) {
    imprimirTabuleiro(tabuleiro);
    simbolo = (jogador == 1) ? 'X' : 'O';
    printf("Jogador %d (%c), digite linha e coluna (0, 1  
ou 2): ", jogador, simbolo);
    scanf("%d %d", &linha, &coluna);

    if (linha < 0 || linha > 2 || coluna < 0 || coluna >
2 || tabuleiro[linha][coluna] != ' ') {
        printf("Jogada inválida! Tente novamente.\n");
        continue;
    }

    tabuleiro[linha][coluna] = simbolo;

    if (verificarVitoria(tabuleiro)) {
        imprimirTabuleiro(tabuleiro);
        printf("Jogador %d venceu!\n", jogador);
        break;
    }

    if (tabuleiroCheio(tabuleiro)) {
        imprimirTabuleiro(tabuleiro);
        printf("Empate!\n");
        break;
    }

    jogador = (jogador == 1) ? 2 : 1;
}
```

```
    return 0;  
}
```