17-02-2024

Jorge Fernandes

ISTEC

Jogo da Velha

Introdução à Programação

Índice

[Introdução 1](#_Toc190362204)

[Descrição do Problema 1](#_Toc190362205)

[Funcionalidades Implementadas 1](#_Toc190362206)

[Implementação 1](#_Toc190362207)

[Ambiente de Desenvolvimento 1](#_Toc190362208)

[Organização do Código 2](#_Toc190362209)

[Principais Desafios 2](#_Toc190362210)

[Testes 2](#_Toc190362211)

[Conclusão 3](#_Toc190362212)

[Código – Fonte 3](#_Toc190362213)

## Índice de Imagens

[Figura 1 - Bibliotecas 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362235)

[Figura 2 – inicializarTabuleiro() 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362236)

[Figura 3 - mostrarTabuleiro() 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362237)

[Figura 4 - verificarVitoria() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362238)

[Figura 5 - verificarEmpate() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362239)

[Figura 6 - jogadaValida() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362240)

[Figura 7 - jogar() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362241)

# Introdução

O objetivo deste trabalho foi desenvolver um jogo da velha (tic-tac-toe) na linguagem C, permitindo que dois jogadores possam interagir em um tabuleiro 3x3, alternando as suas jogadas até que haja um vencedor ou um empate.

# Descrição do Problema

O jogo da velha é um jogo clássico de dois jogadores, onde cada um alterna entre os símbolos 'X' e 'O' em um tabuleiro 3x3. O sistema deve garantir que apenas jogadas válidas sejam realizadas e detetar automaticamente o vencedor ou um empate.

# Funcionalidades Implementadas

* Exibição do tabuleiro atualizado a cada jogada.
* Alternância entre os jogadores.
* Validação de jogadas para impedir entradas inválidas.
* Deteção de vitória ou empate.
* Possibilidade de reiniciar o jogo após o término.

# Implementação

## Ambiente de Desenvolvimento

* Linguagem de programação utilizada: C
* Ferramentas: Dev-C++ (IDE)
* Ambiente: Windows

# Organização do Código

O código foi dividido em funções principais para garantir modularidade e organização:

* inicializarTabuleiro(): Prepara o tabuleiro para uma nova partida.
* mostrarTabuleiro(): Exibe o estado atual do tabuleiro.
* jogadaValida(): Verifica se a jogada escolhida é válida.
* verificarVitoria(): Analisa se algum jogador venceu.
* verificarEmpate(): Confirma se o jogo terminou empatado.
* jogar(): Controla o fluxo principal do jogo.

# Principais Desafios

* Implementar a validação de entrada para impedir loops infinitos.
* Garantir a deteção correta de vitórias e empates.
* Melhorar a usabilidade para que os jogadores tenham uma experiência fluida.

# Testes

Foram realizados vários testes para garantir o funcionamento correto do jogo, incluindo:

* Testes de inserção de jogadas inválidas.
* Verificação automática de vitórias nas linhas, colunas e diagonais.
* Simulação de partidas completas para testar a deteção de empate.

Todos os testes foram concluídos com sucesso, demonstrando que o sistema é capaz de lidar corretamente com todas as regras do jogo.

# Conclusão

O projeto do jogo da velha foi concluído com sucesso e atendeu aos seus objetivos. Ele permite que dois jogadores interajam de forma intuitiva e justa. Durante o desenvolvimento, foi possível aprimorar conhecimentos sobre código em C.

# Código – Fonte

Define o tamanho do tabuleiro como 3x3 e declara uma matriz 3x3 para armazenar o estado do tabuleiro.

Figura 1 - Bibliotecas

Preenche todas as posições da matriz tabuleiro com ' ', indicando que todas as casas estão vazias.

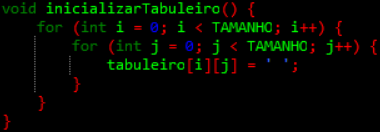


Figura 2 – inicializarTabuleiro()

Cria o tabuleiro.

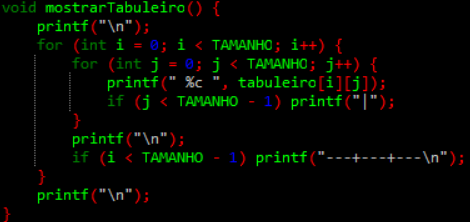


Figura 3 - mostrarTabuleiro()

Verifica se um jogador venceu, verificando:

* Linhas: Se todos os três elementos de alguma linha são iguais.
* Colunas: Se todos os três elementos de alguma coluna são iguais.
* Diagonais: Se os três elementos de uma diagonal são iguais.

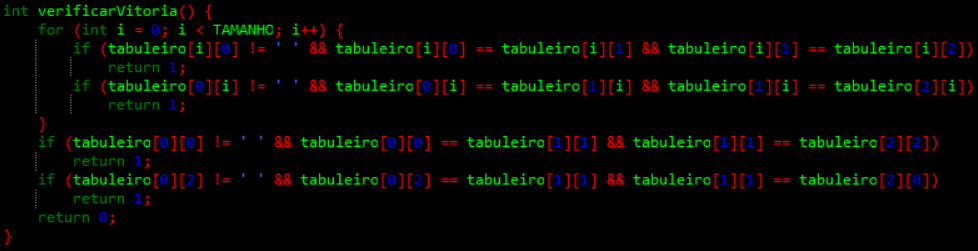
Retorna 1 (true) se houver um vencedor, caso contrário retorna 0.

Figura 4 - verificarVitoria()

Percorre toda a matriz tabuleiro e se houver alguma célula vazia, retorna 0 (não é empate).

Se todas as células estiverem preenchidas e ninguém venceu, retorna 1, ou seja, empate.

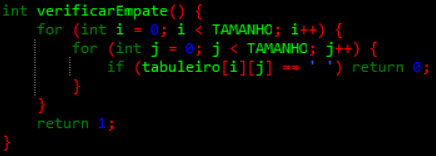


Figura 5 - verificarEmpate()

Verifica se a posição escolhida está dentro dos limites do tabuleiro e se está vazia.

Figura 6 - jogadaValida()

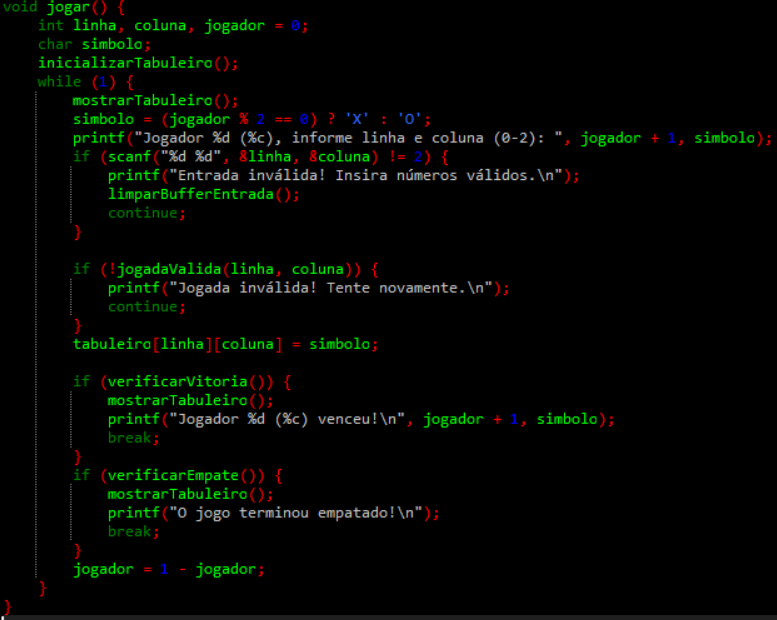
O jogo começa por inicializar o tabuleiro e, em seguida, solicita que o jogador atual (X ou O) faça a sua jogada. Após receber a entrada, verifica se a jogada é válida; caso contrário, exibe uma mensagem de erro e solicita uma nova tentativa. Se a jogada for válida, o tabuleiro é atualizado com o símbolo correspondente. Em seguida, o sistema verifica se há um vencedor e finaliza o jogo se necessário. Se todas as posições estiverem preenchidas e nenhum jogador tiver vencido, a partida é encerrada como empate.

Figura 7 - jogar()