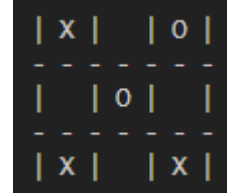


Estrutura de dados - Trabalho Substitutivo

Nome: Gabriel P Toledo

Matrícula:21195

Documentação do trabalho



Introdução:

O software em questão é um jogo de tabuleiro em que dois jogadores preenchem posições em um tabuleiro de 3 por 3. O objetivo é formar uma linha, coluna ou diagonal que tenha o mesmo símbolo (X ou O) antes do oponente.

Uma visão geral da forma como o programa funciona:

Os arquivos "tabuleiro.h", "tabuleiro.c" e "jogo.c" constituem os componentes do programa.

O arquivo "tabuleiro.h" contém as definições das funções que são usadas para alterar o tabuleiro do jogo. Ele define as funções abaixo:

- escreverTabuleiro: escreve o tabuleiro na tela.
- preencherTabuleiro: preenche uma posição do tabuleiro com o símbolo do jogador atual.
- verificarVitoria: verifica se o jogador atual venceu o jogo.
- verificarEmpate: verifica se o jogo terminou em empate.
- jogarNovamente: pergunta ao jogador se deseja jogar novamente.

O arquivo "tabuleiro.c" contém a implementação dessas funções definidas em "tabuleiro.h". Elas são responsáveis por realizar as operações necessárias no tabuleiro.

O arquivo "jogo.c" é o programa principal. Ele utiliza as funções do arquivo "tabuleiro.h" para implementar a lógica do jogo. Neste arquivo, há um loop principal que controla o fluxo do jogo. Ele solicita a entrada dos jogadores, valida as jogadas, verifica vitórias e empates, e permite que os jogadores joguem novamente.

Implementação:

Arquivo "tabuleiro.h":

Define as funções necessárias para manipular o tabuleiro do jogo.

- A função `escreverTabuleiro` recebe a matriz do tabuleiro e imprime seu conteúdo na tela.
- A função `preencherTabuleiro` recebe a matriz do tabuleiro, uma linha, uma coluna e o símbolo do jogador atual. Ela preenche a posição especificada com o símbolo do jogador.
- A função `verificarVitoria` recebe a matriz do tabuleiro e o símbolo do jogador atual. Ela verifica se o jogador venceu o jogo, verificando as linhas, colunas e diagonais.
- A função `verificarEmpate` recebe a matriz do tabuleiro e verifica se o jogo terminou em empate, ou seja, se todas as posições estão preenchidas.
- A função `jogarNovamente` pergunta ao jogador se deseja jogar novamente e retorna 1 se a resposta for afirmativa (S ou s), ou 0 caso contrário.

Arquivo "tabuleiro.c":

- Implementa as funções definidas no arquivo "tabuleiro.h".
- A função `escreverTabuleiro` percorre a matriz do tabuleiro e imprime seu conteúdo na tela, formatando-o adequadamente.
- A função `preencherTabuleiro` recebe a matriz do tabuleiro, a linha e coluna desejadas e o símbolo do jogador atual. Ela atualiza a posição do tabuleiro com o símbolo do jogador.

- A função `verificarVitoria` percorre as linhas, colunas e diagonais do tabuleiro, verificando se alguma delas contém o símbolo do jogador atual em todas as posições. Se isso ocorrer, ela retorna 1 para indicar a vitória.
- A função `verificarEmpate` percorre todas as posições do tabuleiro e verifica se alguma delas ainda está vazia. Se todas as posições estiverem preenchidas, ela retorna 1 para indicar o empate.
- A função `jogarNovamente` solicita ao jogador que escolha se deseja jogar novamente, lendo a resposta do terminal. Ela retorna 1 se o jogador escolher jogar novamente (S ou N), ou 0 caso contrário.

Arquivo "jogo.c":

- É o programa principal que implementa a lógica do jogo.
- O programa utiliza um loop principal para controlar o fluxo do jogo.
- No início do loop, um tabuleiro vazio é criado, representado por uma matriz 3x3 preenchida com espaços em branco.
- O tabuleiro é exibido na tela usando a função `escreverTabuleiro`.
- O primeiro jogador é solicitado a escolher seu símbolo (X ou O) e a escolha é validada.
- Com base na escolha do primeiro jogador, o símbolo do segundo jogador é determinado.
- Dentro de outro loop, as jogadas dos jogadores são solicitadas e validadas.

Conclusão:

No geral, o trabalho foi concluído com sucesso. O programa executa o jogo de forma funcional, permitindo que dois jogadores joguem um contra o outro, preenchendo posições no tabuleiro e decidindo quem ganhou ou perdeu. Outra dificuldade encontrada durante a implementação foi lidar com a validação das entradas do jogador. Garantir que as entradas de linha, coluna e escolha do símbolo do jogador sejam válidas, dentro dos limites do tabuleiro e com os caracteres.

Bibliografia:

Foram utilizados 2 repositórios como base para esse programa um deles foi feito por mim em python em 2022.

<https://github.com/guilhermeow/Dynzch>

<https://github.com/GabrielToledo01/Jogo-da-velha>

Site utilizado para tirar dúvidas de syntax e estrutura da linguagem C:

<https://www.freecodecamp.org/portuguese/news/o-manual-do-iniciante-em-c-aprenda-o-basico-sobre-a-linguagem-de-programacao-c-em-apenas-algumas-horas/>