Trabalho de Introdução a programação

By: Victor Manoel Silva Nascimento, Victor Manuel da Silva Nascimento e Kassandra de Souza Silva

Jogo de dama

O jogo da dama permite que seja jogado por dois jogadores, as peças são identificadas por **X** | **0**. Ganha a partida o jogador quer eliminar todas as peças do seu rival. A movimentação das peças foi inspirada no modo de jogo de batalha naval.

O jogador é instruído a selecionar primeiramente uma linha e uma coluna em que alguma de suas peças está localizada, e ao escolher a peça faz a mesmo para movimentá-la, assim ela pode se deslocar várias casas na diagonal.

Diferente do jogo de dama convencional, dentro do jogo não existe a regra em que a peça ao chegar na 8 linha do tabuleiro é promovida à DAMA.

Também surgindo como uma regra nova, ao existir a possibilidade de comer uma peça do adversário na posição contrária à do jogo oficial, é permitido fazer a jogada.

Se houver uma peça de seu adversário com a possibilidade de ser eliminada, e o jogador não realizar a jogada que causaria sua eliminação ele será penalizado perdendo a peça que deveria realizar a eliminação da peça de seu adversário.

Funções do codigo

- pauseExecution(int milliseconds/seconds):

Esta função pausa a execução do programa por um tempo especificado, usando funções diferentes dependendo do sistema operacional:
No Windows (_WIN32), utiliza a função Sleep(milliseconds) da API do Windows.
Em sistemas UNIX-like, utiliza a função sleep(seconds) da biblioteca unistd.h.
Ela não retorna nenhum valor, apenas pausa a execução por um período determinado.

clear():

A função clear() é responsável por limpar o console, utilizando códigos ANSI específicos (\033[2J\033[H) que são interpretados pelo terminal para limpar a tela e posicionar o cursor no topo (home).
Ela não retorna valor algum; sua função é apenas realizar a limpeza do console.

- resetCheckers(int ch[CHECKERSLEN][CHECKERSLEN]):

Esta função recebe como parâmetro uma matriz ch que representa o tabuleiro de damas. Ela redefine todas as posições da matriz para zero, indicando que não há peças no tabuleiro.
Não retorna nenhum valor.

- verifyValidMove(int cp, int jp, int cp2, int jp2, int ch[CHECKERSLEN] [CHECKERSLEN]):

A função verifyValidMove() verifica se um movimento de uma peça de coordenadas (cp, jp) para (cp2, jp2) é válido dentro das regras do jogo de damas.
Retorna um valor inteiro:
1 se o movimento é válido.
0 se o movimento não é válido.
Testa se:
As coordenadas estão dentro dos limites do tabuleiro.
A peça selecionada pode mover-se na direção correta.
Se for uma dama, verifica também se pode capturar uma peça adversária.

- canMovePiece(int cp, int jp, int ch[CHECKERSLEN][CHECKERSLEN], int whoseTurn):

Verifica se a peça nas coordenadas (cp, jp) pode ser movida pelo jogador atual (whoseTurn).
Retorna um valor inteiro:
1 se a peça pode ser movida.
0 se a peça não pode ser movida.
Testa:
Se as coordenadas (cp, jp) estão dentro dos limites do tabuleiro.
Se a peça pertence ao jogador atual (whoseTurn).

Se há movimentos válidos disponíveis para essa peça.

- checkWinner(int ch[CHECKERSLEN][CHECKERSLEN], int *winner, int whoseTurn):

Verifica se há um vencedor no jogo de damas.
Retorna um valor inteiro:
1 se há um vencedor.
0 se não há um vencedor ainda.
Altera o valor apontado por *winner para indicar o jogador vencedor (1 para X, 2 para O, 0 para empate).
Testa:
Se um dos jogadores não tem mais movimentos possíveis.
Se todas as peças de um jogador foram capturadas.
Calcula o número de peças de cada jogador para determinar o vencedor.

- displayCheckers(int ch[CHECKERSLEN][CHECKERSLEN]):

Exibe o estado atual do tabuleiro de damas.
Não retorna valor algum; apenas imprime o tabuleiro no console.
Utiliza códigos de formatação simples para representar as peças (X para jogador 1, O para jogador 2) e espaços vazios.
capturePiece(int cp, int jp, int cp2, int jp2, int

- capturePiece(int cp, int jp, int cp2, int jp2, int ch[CHECKERSLEN][CHECKERSLEN]):

Verifica se uma peça pode capturar outra.
Retorna um valor inteiro:
1 se houve captura.
0 se não houve captura.
Remove a peça capturada se o movimento resultar em uma captura.

- checkCaptures(int cp, int jp, int ch[CHECKERSLEN][CHECKERSLEN]):

Verifica se uma peça pode capturar outra, verificando as quatro possíveis direções de captura.
Retorna um valor inteiro:
1 se há pelo menos uma captura possível.
0 se não há capturas possíveis.

canCapturePiece(int player, int c2, int j2, int

- canCapturePiece(int player, int c2, int j2, int ch[CHECKERSLEN] [CHECKERSLEN]):

Verifique se uma peça pode capturar outra, se não houver outra opção.
Remova a peça se não houver opções de captura.
Não retorna valor, apenas atualiza o tabuleiro se necessário.

- makeMove(int ch[CHECKERSLEN][CHECKERSLEN], int *whoseTurn):

Coordena a lógica para permitir que um jogador faça um movimento válido no jogo de damas.
Não retorna valor; controla diretamente o fluxo do jogo.
Interage com o jogador para obter a peça a ser movida e sua posição de destino, realizando a movimentação no tabuleiro e verificando capturas.

- fillCheckers(int ch[CHECKERSLEN][CHECKERSLEN]):

Preencher o tabuleiro de damas com as peças nas posições iniciais.
Não retorna valor, apenas atualiza a matriz ch que representa o tabuleiro.

- gameloop():

Implementa o loop principal do jogo de damas.
Não retorna valor; controla diretamente o fluxo do jogo.
Chama as funções necessárias para preencher o tabuleiro, permitir movimentos de jogadores, verificar vitórias e exibir o estado do jogo.
main():

Função principal que inicia o jogo.
Retorna 0 ao finalizar o programa (padrão de finalização do programa em C).
Chama gameloop() para iniciar o jogo de damas.