

**Licenciatura Engenharia Informática**

**Programação 2020/2021**

**Jogo do Semáforo**



O jogo do semáforo: um jogo entre 2 pessoas que efetuam jogadas alternadas, até que uma delas vença ou que se verifique um empate.

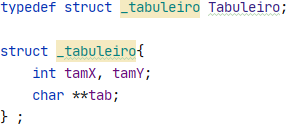
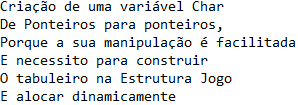
Ricardo Ferreira – 21260251

1. **Introdução**

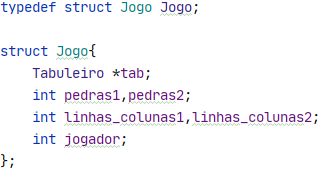
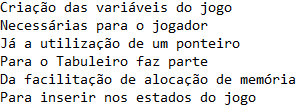
O jogo do semáforo desenrola-se num tabuleiro dividido em células. No início, o tabuleiro está vazio. Alternadamente os jogadores vão colocando peças de cor Verde (G), Amarela (Y) ou Vermelha (R). Ganha o jogador que coloque uma peça que permita formar uma linha, coluna ou diagonal completa1 com peças da mesma cor.

1. **Estruturas Principais**

**Tabuleiro**

**Jogo**

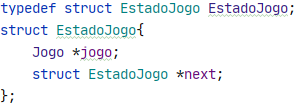
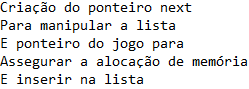
 

**EstadoJogo**



Ponteiro para o início da lista com o propósito de assegurar e impedir a perda do ponteiro inicial da lista.

Número dos estados de jogo serve garantir a gravação, exportação e importação dos vários estados de jogo.

1. **Módulos Principais**

Efetivamente, o código está estruturado em cinco módulos, que são estes respetivamente, Interface.h, Estadojogo.h, Jogo.h, Tabuleiro.h.

Interface.h - Contém o menu do jogo com opções.

- Executa comandos do utilizador.

- Imprime Resultado do jogo.

Estadojogo.h - Inicia e guarda o estado do jogo inicial na Lista.

**-** Insere estado do jogo atual na Lista.

**-** Exporta/Importa os estados do jogo para um ficheiro.

Jogo.h - Inicia e guarda o jogo inicial.

- Manipula as opções e variáveis do jogo, verificando se a opção é válida, e consequentemente incrementa as variáveis do jogador atualmente em jogo.

- Liberta memória reservada para o jogo.

Tabuleiro.h - Inicia e guarda o tabuleiro de jogo inicial.

- Duplica o tabuleiro para guardar na lista.

- Liberta memória reservada para o tabuleiro do jogo.

1. **Funcionamento**

Interface.c



Executa e valida o comando dependente da peça que o utilizador atual quer jogar.



Permite ao jogador atual escolher jogar contra o jogador automático e valida as opções do jogo.

Estadojogo.c

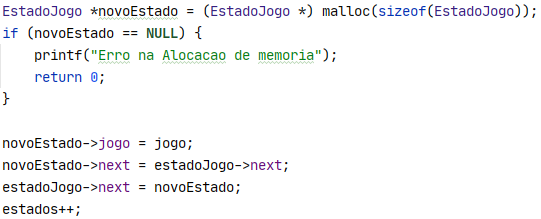


Como este ficheiro contém a lista, o iniciaEstado reserva memória para lista e guarda o estadoJogo de referência (Dummy Node) no início.



O insereEstado está correlacionado com o salvaJogo que reserva memória para jogo ser reservado na lista, isto acontece sempre que há uma alteração no jogo e consequentemente no estadoJogo, incrementando o número de estados do jogo.

Efetivamente, o estadoJogo é inserido no início da lista, porque como uma das opções do jogo é mostrar N estados do jogo, concluí que seria mais fácil que o estado recentemente inserido fosse o último.





Exporta os estados do jogo para um ficheiro binário e de texto, criando o ficheiro após a leitura do seu nome.

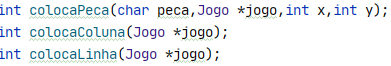


Retorna o jogo gravado no ficheiro binário exportado, reservando memória para todos os estados do jogo anteriormente gravados no ficheiro.

Jogo.c



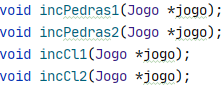
Reserva memória para o jogo inicial e retorna-o.



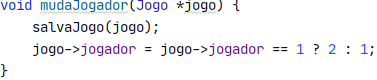
colocaPeca: Primeiramente recebe as coordenadas que o utilizador introduziu e verifica se são válidas, em seguida valida as diferentes pecas, ou seja, se a peça é ‘G’ e o tabuleiro ‘-’ ou a peça é ‘Y’ e o tabuleiro ‘G’ ou a peça é ‘R’ e o tabuleiro ‘Y’ retorna 1, senão retorna 0.

colocaColuna: A função que insere a coluna, garante esta adição através de um ciclo que percorre o número de linhas do tabuleiro e realoca com o uso de um ponteiro auxiliar um espaço novo para essa linha, criando uma coluna. Para além disto, verifica se realocação foi bem sucedida e retorna 1, senão retorna 0. No final atualiza o tamanhoY do tabuleiro do jogo.

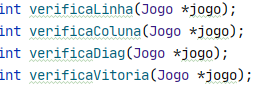
colocaLinha: A função que insere uma linha, primeiro faz uso de um auxiliar de carateres de ponteiros para ponteiros, porque para inserir uma linha é necessário realocar espaço para uma nova referência do tamanhoX do Tabuleiro e em seguida aloca-se memória para a nova linha. Para além disto, verifica se realocação foi bem sucedida e retorna 1, senão retorna 0. No final atualiza o tamanhoX do tabuleiro do jogo.



Incrementa variáveis do jogo.



Guarda o jogo, inserindo um novo estado e muda o jogador.



Todas estas funções, exceto o verificaVitoria percorrem o Tabuleiro em busca da Win Condition, ou seja, formar uma linha, coluna ou diagonal completa com peças da mesma cor. Efetivamente, a lógica é semelhante entre elas, uma vez que guarda o carater na pos da linha, coluna e diagonal inicial do tabuleiro comparando se não é um ‘\_’ ou um ‘X’ e depois verifica se as restantes posições são iguais.

verificaVitoria: Retorna 1 se ganhou por linha, 2 se ganhou por coluna e 3 se ganhou por diagonal, senão retorna 0.



Escolhe opção do jogador automático e liberta memória do Tabuleiro e do Jogo.

Tabuleiro.h



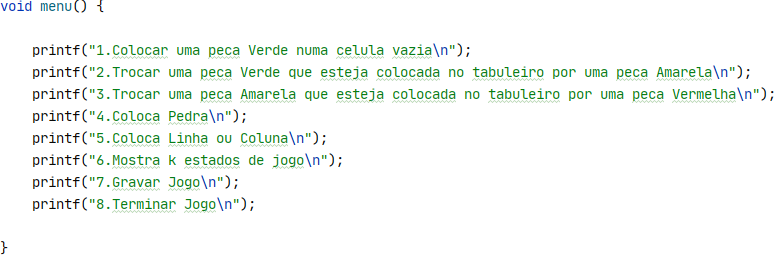
Como a estrutura Tabuleiro contém um vetor de ponteiros para ponteiros, a solução correta para impedir a perda dados é alocar memória, primeiramente para o tabuleiro, depois para a var char de ponteiros e depois para os ponteiros apontados pelos ponteiros. Desta forma a função percorre o tamanho inicializado de forma aleatória na main e coloca ‘-’ nas posições alocadas em memória. Esta função retorna o ponteiro do Tabuleiro criado em memória.



A função que duplica o Tabuleiro faz uso da mesma lógica da criação do principal e serve para aquando a salvação do jogo, não ‘estragar’ o principal.

O freeTabuleiro liberta memória do Tabuleiro, do vetor e dos ponteiros por esse apontado.

1. **Manual do Utilizador**



**As jogadas válidas relativas à colocação de peças são as seguintes:**

1. Colocar uma peça Verde numa célula vazia.

2. Trocar uma peça Verde que esteja colocada no tabuleiro por uma peça Amarela.

3. Trocar uma peça Amarela que esteja colocada no tabuleiro por uma peça Vermelha.

**Existem quatro jogadas adicionais que podem ser efetuadas pelos jogadores:**

4. Colocar uma pedra numa célula vazia. Cada jogador pode colocar, no máximo, uma pedra por jogo. A colocação de uma pedra inviabiliza que o jogo possa terminar por preenchimento da linha e coluna afetadas (e, eventualmente também da diagonal ou diagonais).

5. Adicionar uma linha ou uma coluna ao final do tabuleiro. Esta jogada adiciona linhas ou colunas completas e vazias ao tabuleiro de jogo. Cada jogador pode efetuar esta jogada, no máximo, duas vezes por jogo.

6. Escolher o número de jogadas que quer rever.

7. Gravar o jogo, para depois voltar a jogar mais tarde.