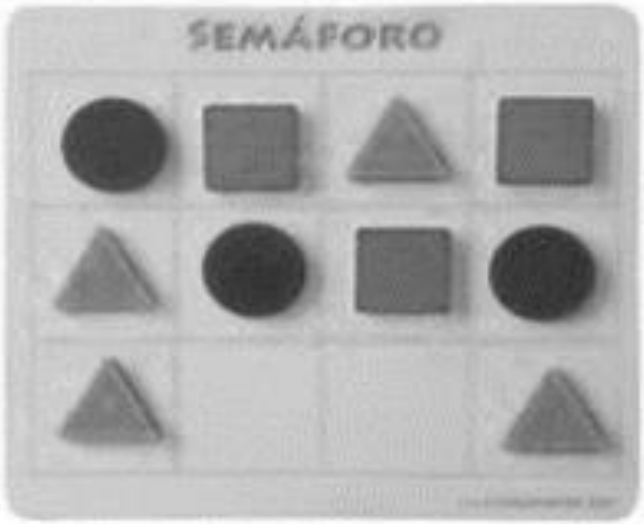
Instituto Superior de Engenharia de Coimbra

Programação
2020/21 | LEICE

Jogo do Semáforo



Docente: Francisco Pereira
Discente: Pedro Praça 2020130980

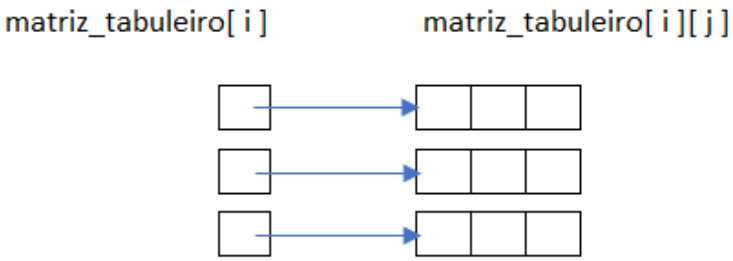
Índice

1.0 Enquadramento

O presente relatório surge no âmbito da criação do jogo semáforo em linguagem c, enquadrando-se na disciplina de Programação.
Ao longo do relatório são apresentados os aspetos mais importantes de implementação do programa, bem como um pequeno manual de utilização.

2.0 Principais Estruturas de Dados

Neste programa, utilizei as seguintes estruturas: uma para guardar o tabuleiro, ou seja, as peças nas posições em que são jogadas; e outra para registar as jogadas efetuadas.
No que toca à primeira, consiste numa matriz alocada/realocada em memória, do tipo char (char **matriz_tabuleiro), a qual se pode representar pela seguinte imagem:



Aquando do início do jogo, a dimensão do tabuleiro é estabelecida aleatoriamente, entre 3x3 e 5x5, sempre *quadrada*. Aí, é alocada a matriz, tal como referido acima. Sempre que um jogador acrescenta uma linha ou uma coluna, é feito o seu redimensionamento, realocando memória.
A segunda estrutura consiste numa lista ligada simples, dinâmica, em que cada nó (*node*) consiste numa *struct*, com o objetivo de registar todas as jogadas efetuadas, uma em cada nó.

A *struct* que armazena os dados de uma jogada tem a seguinte estrutura:

```
typedef struct struct_jogada {
    int num_jogada;     //número da jogada
    int jogador;        //jogador
    char opcao_jogada;  //tipo de jogada
    int linha_jogada;   //linha da jogada (qd inserida)
    int coluna_jogada;  //coluna da jogada (qd inserida)
    int nlinhas_tabuleiro;  //numero de linhas do tabuleiro
    int ncolunas_tabuleiro;     //numero de colunas do tabuleiro
    struct struct_jogada *proximo; //ponteiro que aponta para a proxima struct
} sjogada;
```

As variáveis que compõem a *struct* permite descrever exatamente cada uma das jogadas.
Assim, torna possível reconstituir o tabuleiro em qualquer momento do jogo. Tinha considerado uma outra possibilidade, que seria armazenar o *tabuleiro* numa matriz dinâmica em cada um dos nós da lista, facilitando a apresentação das jogadas. Contudo, como ocuparia demasiada memória, optei por esta solução e implementei uma função que reconstitui o tabuleiro, jogada a jogada, apresentando apenas as ***n*** que o utilizador pede.

3.0 Pormenores da implementação

Apresento abaixo uma lista com alguns pormenores da implementação que considero merecerem destaque:

- Criei um ficheiro apenas com as funções que lidam com o tabuleiro de jogo (semaforo.c e semaforo.h), desde a criação da matriz em memória até às funções que verificam a validade das jogadas, etc;
- Utilizei o ficheiro fornecido pelo Professor (utils.c e utils.h), com as funções para gerar números aleatórios, ao qual acrescentei uma função para limpar o ecrã (limpa_ecran);
- No ficheiro main.c:
    - faço o *include* dos ficheiros referidos;
    - defini a *struct* que regista uma jogada e permite criar a lista ligada;
    - criei todas as funções que manipulam a lista ligada.
    - Nota: não consegui colocar estas funções num ficheiro separado, pois a struct não era reconhecida. Por isso, optei por colocar no main.

4.0 Manual de Utilização

4.1 Interface

O programa do jogo semáforo apresenta uma interface de intuitivo e de utilização simples. Logo no início é nos apresenta o tabuleiro quadrado cuja dimensão entre 3 e 5 linhas, o valor é selecionado aleatoriamente. O tabuleiro é iniciado por espaços em brancos indicados pelo "_" seguido da legenda de cada opção de jogada que o jogador pode fazer, assim como a vez de cada jogador.

```
*** Jogo do semaforo ***

   1  2  3  4
  ------------
1| _  _  _  _
2| _  _  _  _
3| _  _  _  _
4| _  _  _  _
```

Figura 1 -Tabuleiro

```
Opcoes:
   G: peca verde
   Y: peca amarela
   R: peca vermelha
   P: pedra
   C: acrescentar coluna
   L: acrescentar linha

   V: ver ultimas jogadas

   T: terminar

Indique a sua jogada, jogador 1:
```

Figura 2 -Legenda das opções

4.2 Jogadas

Escolhendo a opção G, por exemplo, estamos a escolher por uma peça Verde numa posição do tabuleiro, logo de seguida são pedidas as coordenadas para colocar a peça, primeiro a linha e de seguida a coluna.

```
Indique a sua jogada, jogador 1: g

Indique a linha [1 - 4]: 2

Indique a coluna [1 - 4]: 3
```

Figura 3 -Indicação da linha e coluna

```
   1  2  3  4
  ------------
1| _  _  _  _
2| _  _  G  _
3| _  _  _  _
4| _  _  _  _
```

Figura 4 -Peça G nas posições indicadas

A seguir às opções de colocar peças no tabuleiro temos duas opções que só podemos usufruir no total duas vezes, que são as opções de acrescentar uma linha ou uma acrescentar uma coluna ao tabuleiro.

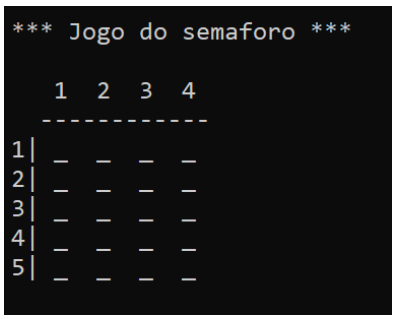


Figura 5 -Acrescentar linha

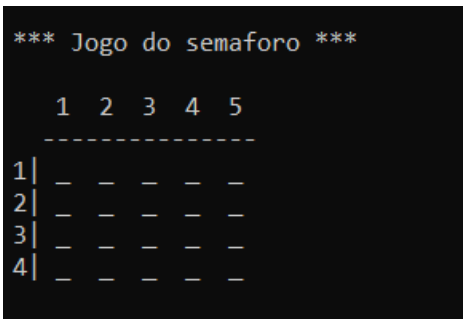


Figura 6 -Acrescentar coluna

Por último temos a opção de jogar uma pedra (“o”), que só se pode fazer esta jogada uma vez durante a durante o jogo. Nesta jogada também são pedidas as coordenadas para colocar a pedra, como nas peças verdes, vermelhas e amarelas.

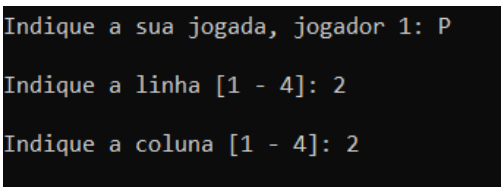


Figura 7 -Indicação da linha e coluna

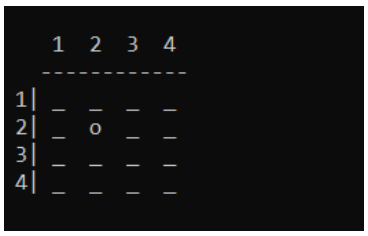
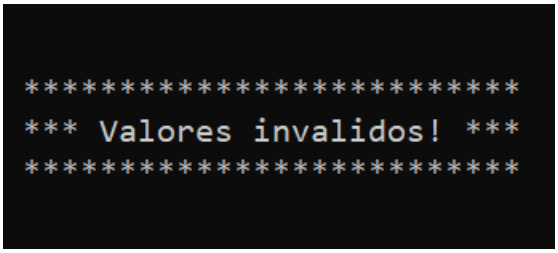


Figura 8-Pedra posicionada nas coordenadas indicadas

O programa é capaz de detetar se as coordenadas indicadas são inválidas e até mesmo se opção “pedra” ou as opções de “acrescentar linha” ou “acrescentar coluna” já foram usadas dentro dos limites definidos. Nestes casos é apresentada a mensagem seguinte:



O utilizador também tem a opção de ver a *n* jogadas anteriores através da seguinte opção:

```
V: ver ultimas jogadas
```

De seguida é pedido as *n* jogadas que o utilizador que ver, após esse paço são apresentadas as *n* jogadas. Por exemplo, se o utilizador decidir visualizar as últimas 3 jogadas:

```
*** Jogada # 3 ***

   1  2  3  4
  ------------
1| G  _  _  _
2| _  Y  _  _
3| _  _  _  _
4| _  _  _  _

*** Jogada # 4 ***

   1  2  3  4
  ------------
1| G  _  _  _
2| G  Y  _  _
3| _  _  _  _
4| _  _  _  _

*** Jogada # 5 ***

   1  2  3  4
  ------------
1| G  _  _  _
2| G  Y  _  _
3| _  _  _  _
4| _  _  _  G
```

```
Indique a sua jogada, jogador 2: v

Indique quantas jogadas quer visualizar [1 - 5]: 3
```

5.0 Conclusão

Devido à coincidência da realização de trabalhos de várias disciplinas, e do tempo que a investigação para este trabalho tomou, não consegui implementar o *jogador automático* nem a interrupção/retoma do jogo (gravação e carregamento das jogadas já realizadas).

Contudo, penso que as funcionalidades ficaram bem estruturadas e sólidas e com uma interface muito simples e intuitiva assegurando uma boa experiência a quem for experimentar o jogo.