

Departamento de Engenharia Informática e de Sistemas

Instituto Superior de Engenharia de Coimbra

Licenciatura em Engenharia Informática

Programação - 2020/2021

Relatório do Trabalho Prático:

Jogo do Semáforo

Turma Prática Nº 4

André Lopes | a2019139754@isec.pt

Índice

1. Introdução	3
2. Estruturas de Dados	4
3. Estruturas Dinâmicas.....	6
4. Justificação para as opções tomadas em termos de implementação	8
5. Manual de Utilização.....	10
6. Conclusão	13
7. Bibliografia	14

1. Introdução

No âmbito da Unidade Curricular de Programação, pretendeu-se desenvolver o jogo do semáforo, de acordo com as regras que foram estabelecidas. O programa foi implementado na linguagem C standard, respeitando a norma C99, e foi usado o Clion como ambiente de desenvolvimento.

Ao longo do relatório, vão ser explicadas os conceitos da linguagem usados tais como as estruturas de dados, estruturas ligadas, a leitura e escrita de ficheiros, que foram necessários para o respetivo funcionamento do jogo.

2. Estruturas de Dados

Para a implementação deste trabalho, foram utilizadas 3 estruturas de dados diferentes que se encontram declaradas no ficheiro dados.h.

1. Estrutura Jogador

```
typedef struct jogador Jogador, *pJogador;  
struct jogador{  
    char nome[20];  
    int pedras;  
    int linCol;  
};
```

A estrutura jogador é usada para guardar a informação relativas aos jogadores, mais concretamente o seu nome, o número de pedras que pode jogar e o número de linhas/colunas que pode adicionar. Estes dados jogadores vão ser guardados num array de estruturas declarado no ficheiro jogo.c.

```
Jogador jogadores[2];
```

2. Estrutura Tabuleiro:

```
typedef struct tabuleiro Tabuleiro, *pTabuleiro;  
struct tabuleiro{  
  
    int nLinhas;  
    int nColunas;  
    char **Tab;  
};
```

A estrutura tabuleiro é usada para guardar a informação relativa ao Tabuleiro que está sempre a sofrer alterações. Este apresenta o número de linhas e colunas que o tabuleiro tem, assim como um array bidimensional do tipo char.

3. Estrutura InfoJogada:

```
typedef struct infoJogada InfoJogada, *pInfoJogada;  
struct infoJogada{  
  
    int jogadorJogar;  
    int numJogada;  
    int modo;  
    pTabuleiro tabuleiro;  
    pInfoJogada prox;  
};
```

A estrutura infoJogada é usada para guardar a informação relativa a cada jogada do jogo. Esta apresenta o jogador a jogar, o número da jogada, o modo de jogo escolhido, um ponteiro que aponta para a estrutura referente ao tabuleiro, assim como o ponteiro prox que vai apontar para a próxima jogada da lista ligada.

3. Estruturas Dinâmicas

O tabuleiro foi mantido numa estrutura dinâmica, e foi alocada dinamicamente numa tabela bidimensional de caracteres. É criado a partir da estrutura `infoJogada` que armazena a estrutura tabuleiro. É feita a alocação em memória para essa estrutura e de seguida inicializa-se o tabuleiro.

`void inicializaTabuleiro(pInfoJogada jogada):`

Esta função é responsável primeiramente por alocar memória para a estrutura tabuleiro, por definir a dimensão do array, por alocar a memória para o array dinâmico bidimensional e por alocar memória para cada linha do tabuleiro, para que mais tarde seja possível fazer as alterações.

`void adicionaLinha(pInfoJogada inf, pJogador jogador);`

`void adicionaColuna(pInfoJogada inf, pJogador jogador);`

Estas duas funções são responsáveis por realocar memória para acrescentar novas linhas/colunas ao tabuleiro dinâmico.

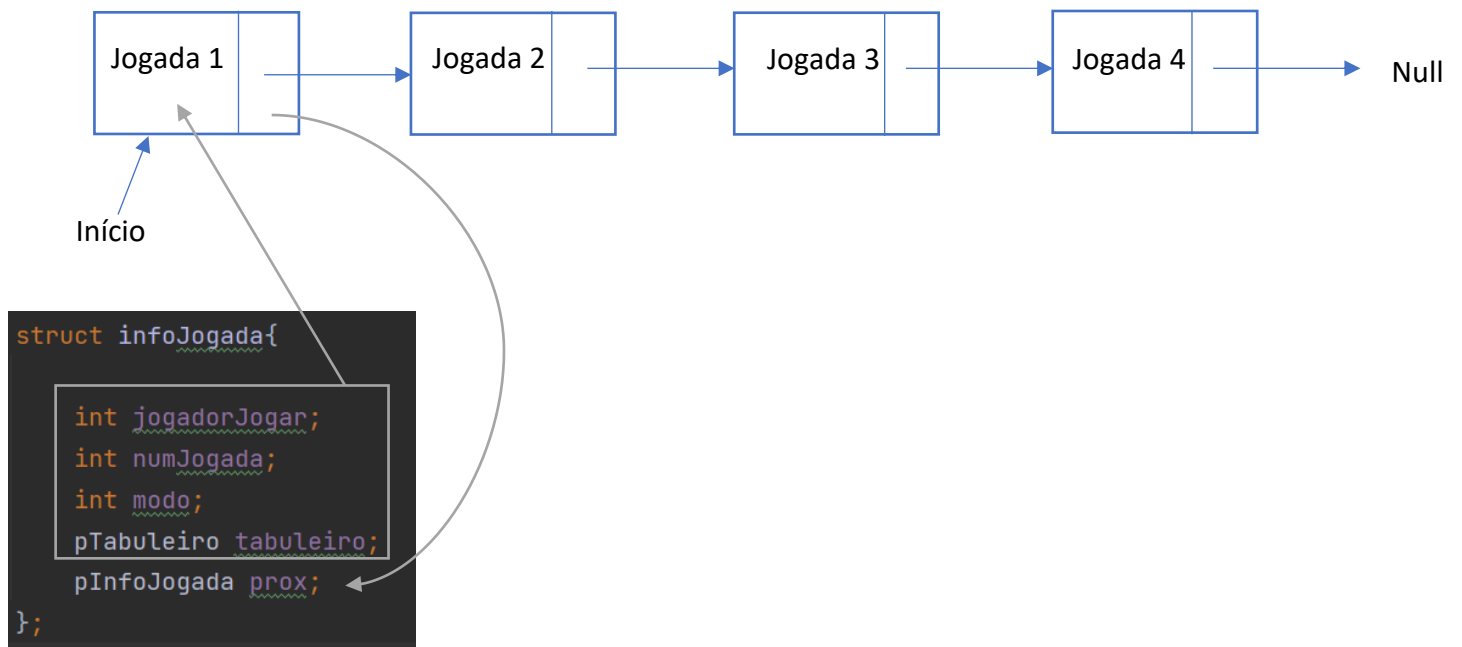
Relativamente às diferentes jogadas que vão ocorrendo durante o jogo, estas foram guardadas numa estrutura dinâmica do tipo lista ligada que foi criada no `jogo.c`

```
pInfoJogada listaJogadas = NULL;
```

`pInfoJogada guardaTabuleiro(pInfoJogada atual, pInfoJogada lista):`

Esta função é responsável por alocar memória para lista caso esta ainda esteja vazia, assim como para todos os elementos pela qual é composta. Caso esta já se encontre com uma jogada, ela vai inserir uma nova jogada no final da lista ligada com os dados recebidos por argumento da jogada atual. A utilidade do `prox`, serve para colocar a próxima jogada a

NULL, e da próxima vez que se guardar uma nova jogada, vai se percorrer a lista até chegarmos ao seu último nó.



4. Justificação para as opções tomadas em termos de implementação

Em termos de implementação do jogo, no ficheiro jogo.c é onde estão presentes todas as funções relativas ao funcionamento do programa e a verificações do jogo, e no jogo.h as suas declarações.

Isto leva em conta, a verificação da peça jogada tanto pelo jogador humano, como pelo computador que é sorteada, assim como as verificações relativas a uma linha/coluna/diagonal completa que por sua vez, indicam o fim do jogo.

```
int jogada(pInfoJogada lista, pInfoJogada inf, Jogador jogadores[]);
void jogarPecaHumano(pInfoJogada inf);
void jogarPecaAutomatico(pInfoJogada inf);
int fimJogo(pInfoJogada inf);

int colunasIguais(pInfoJogada inf, char letra);
int verificaColunas(pInfoJogada inf);
int linhasIguais(pInfoJogada inf, char letra);
int verificaLinhas(pInfoJogada inf);
int diagonaisIguais(pInfoJogada inf, char letra);
int verificaDiagonais(pInfoJogada inf);

void adicionaLinha(pInfoJogada inf, pJogador jogador);
void adicionaColuna(pInfoJogada inf, pJogador jogador);
void jogarPedra(pInfoJogada inf, pJogador jogador);

void mostrarJogadasAnteriores(pInfoJogada lista, int num, int total);
int contaJogadas(pInfoJogada lista);
pInfoJogada guardaTabuleiro(pInfoJogada atual, pInfoJogada lista);
```

No ficheiro dados.c e dados.h, estão presentes as funções responsáveis por inicializar o tabuleiro e os dados relativos aos dois jogadores.

```
//DADOS DO TABULEIRO
void mostraTabuleiro(pInfoJogada info);
void inicializaTabuleiro(pInfoJogada jogada);
void mostraTabuleiroVazio();

//DADOS DOS JOGADORES
Jogador atribuirDadosJogador(char *nome);
void adicionarJogador(Jogador jogador[], char *nome, int id);
void mostrarDadosJogador(Jogador jogador);
```


Relativamente à leitura e escrita de ficheiros, para a gravação da sucessão dos estados do tabuleiro para o ficheiro .txt, foi percorrida a lista ligada com a informação relativa a todas as jogadas e com uso da função `fprintf`, foi escrita essa informação para o ficheiro.

Para a escrita e leitura do ficheiro binário, foram guardadas todas as estruturas fundamentais para a execução do jogo num carregamento. Isto inclui as estruturas dos jogadores, da jogada atual e da lista ligada com informação relativa a outras jogadas, que foram lidas com o `fread` e escritas com o `fwrite`.

```
void gravaJogoTexto(pInfoJogada lista);  
void gravarJogo(pInfoJogada info, pInfoJogada lista, Jogador  
*jogadores);  
void carregarJogo(pInfoJogada info, pInfoJogada lista, Jogador  
*jogadores);  
int verificarFicheiro();
```

Em relação a toda a memória que é alocada durante o funcionamento do programa, esta vai ser libertada quando ocorre o final do jogo.

```
void libertarMemoria(pInfoJogada jogada, pInfoJogada todasJogadas){  
    //Libertar memoria  
    for (int i = 0; i < jogada->tabuleiro->nLinhas; i++)  
        free(jogada->tabuleiro->Tab[i]);  
  
    free(jogada->tabuleiro);  
    free(jogada);  
  
    pInfoJogada aux = todasJogadas;  
  
    while(aux != NULL){  
        aux = aux->prox;  
        for (int i = 0; i < todasJogadas->tabuleiro->nLinhas; i++)  
            free(todasJogadas->tabuleiro->Tab[i]);  
  
        free(todasJogadas->tabuleiro);  
        free(todasJogadas);  
        todasJogadas = aux;  
    }  
}
```

5. Manual de Utilização

Assim que o jogo começa é apresentado um menu com 3 opções, pode iniciar um novo jogo, visualizar as regras do jogo ou sair do jogo, e é pedido ao utilizador para seleccionar uma.

```
----- Jogo do Semaforo -----  
  
      - - - - -  
      - - - - -  
      - - - - -  
      - - - - -  
      - - - - -  
  
1. Iniciar Jogo  
2. Regras do Jogo  
3. Sair  
  
Opcao:
```

Ao seleccionar iniciar jogo, o programa vai verificar se existe algum ficheiro binário guardado respetivo a algum jogo anterior, e caso existe, vai aparecer o seguinte menu:

```
1. Iniciar Jogo  
2. Regras do Jogo  
3. Sair  
  
Opcao: 1  
  
Deseja continuar o jogo anterior?  
  
1. Sim  
2. Nao  
  
Opcao:
```

Caso o utilizador não queira continuar um jogo anterior, ser-lhe-á solicitado a escolher o modo de jogo que pretende.

```
Deseja continuar o jogo anterior?  
  
1. Sim  
2. Nao  
  
Opcao: 2  
  
Indique o modo de jogo que pretende jogar:  
  
1. Jogador A(Humano) VS Jogador B(Humano)  
2. Jogador A(Humano) VS Computador  
  
Opcao:
```

Ao seleccionar um modo, vai aparecer um menu de jogada onde estão apresentadas informações relativas às jogadas e ao jogador. Para além disso, aparecem-lhe as diferentes opções que pode jogar.

```
----- 1ª Jogada -----  
  
      Tabuleiro do Jogo:  
  
        1  2  3  4  
1      -  -  -  -  
2      -  -  -  -  
3      -  -  -  -  
4      -  -  -  -  
  
[Info] É a vez do Jogador A!  
  
      Pedras: 1 | Lin/Col: 2  
  
1. Colocar peca  
2. Colocar pedra  
3. Adicionar linha/coluna  
4. Visualizar 'K' jogadas anteriores  
5. Guardar o jogo  
  
0. Sair do jogo  
  
Opcao:
```

A partir daqui o funcionamento é sempre o mesmo, tendo o utilizador também a possibilidade de ver jogadas anteriores.

```
1. Colocar peça
2. Colocar pedra
3. Adicionar linha/coluna
4. Visualizar 'K' jogadas anteriores
5. Guardar o jogo

0. Sair do jogo

Opcao: 4

[INFO] Numero de jogadas realizadas: 6

Introduza o numero de jogadas anteriores que pretende visualizar: 2

-----

Jogada 5:

      1 2 3 4
1  Y  _ _ _
2  _  G _ _
3  G  _ 0 _
4  _  _ _ _

Jogada 6:

      1 2 3 4
1  Y  _ _ _
2  _  G _ _
3  G  _ 0 _
4  _  _ _ _
5  _  _ _ _

-----

1. Colocar peça
2. Colocar pedra
3. Adicionar linha/coluna
4. Visualizar 'K' jogadas anteriores
5. Guardar o jogo

0. Sair do jogo

Opcao:
```

Quando se verifica um final do jogo, é pedido ao utilizador para escrever o nome do ficheiro .txt para guardar o registo das sucessões do tabuleiro durante o jogo.

```
Opcao: 1
Indique a posição da peça a jogar (lin,col):
Lin: 5
Col: 2

----- Fim do Jogo -----

Tabuleiro do Jogo:

      1 2 3 4
1  Y  G _ _
2  _  G _ _
3  Y  G 0 _
4  _  G _ _
5  _  G _ _

[Info] O vencedor foi o Jogador A!

Introduza o nome do ficheiro para guardar os dados do jogo: replay

[Info] Informacao relativa ao jogo guardada com sucesso no ficheiro 'replay.txt'!

[Info] Memoria libertada!

Pressione qualquer tecla para sair do jogo...
```

6. Conclusão

Com este trabalho, os conceitos aprendidos na cadeira são mais aprofundados, sendo possível superar algumas dificuldades relativas a esta linguagem de programação.

Umas formas usadas para combater estas dificuldades foi a realização de exercícios de aulas e do uso da ferramenta de debugging.

No fundo, a realização deste trabalho acaba por ser uma ótima forma de aprendizagem da linguagem, assim como de estudo para o exame.

7. Bibliografia

Gookin, D. (s.d.). *C For Dummies*.

King, K. N. (s.d.). *C Programming: A Modern Approach (2nd Edition)*.

Learn C Programming. (s.d.). Obtido de <https://www.programiz.com/c-programming>.

Stackoverflow. (s.d.). Obtido de <https://stackoverflow.com/>