

Trabalho Laboratorial

Programação

Docente: Professor Francisco José Baptista Pereira

Autor:

Daniel Duarte Dias Ferreira Albino - a2020134077 – LEI

Engenharia Informática

Coimbra, junho de 2021

Índice

Índice.....	2
1. Introdução	3
2. Estrutura de dados utilizada.....	4
3. Estrutura dinâmica	5
4. Manual de instruções	8
5. Conclusão	12

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Programação, do curso de Licenciatura Informática do Instituto Superior de Engenharia de Coimbra.

O principal objetivo consiste, na aplicação de conhecimentos sobre a linguagem C, no intuito de desenvolver um programa/jogo proposto.

Com este trabalho procura-se consolidar e aplicar o conhecimento adquirido relativo à linguagem C.

2. Estrutura de dados utilizada

Neste trabalho é usada a seguinte estrutura de dados:

```
typedef struct tabuleiro Itab,*pItab;  
  
struct tabuleiro {  
    int coluna,linha;  
    int ganhou,n_jogadas,termina,jogador,jogadorA,pedra1,pedra2,lc1,lc2;  
    char **tabuleiro;  
    pItab prox;  
};
```

As variáveis coluna e linha armazenam as dimensões do tabuleiro, a variável ganhou armazena 0 ou 1 caso algum dos jogadores ainda não tenha ganho (0) ou se algum já ganhou o jogo (1), a variável n_jogadas armazena as jogadas realizadas, a variável termina armazena 0 ou 1 caso o utilizador pretenda terminar o jogo (1), a variável jogador armazena qual é o jogador a jogar (1 ou 2), a variável jogadorA armazena se o jogo é contra o jogador automático (1) ou se é um jogo de duas pessoas (0), as variáveis pedra1 e pedra2 armazenam as vezes que o jogador 1 ou que o jogador 2 já usaram a peça pedra, as variáveis lc1 e lc2 armazenam as vezes em que o jogador 1 e o jogador 2 já adicionaram uma linha ou uma coluna.

A variável tabuleiro é o tabuleiro do jogo.

Esta estrutura de dados é útil, pois armazena todos os dados do jogo o que facilita a escrita numa lista ligada e/ou ficheiros e são variáveis que são responsáveis pela continuação do jogo (ganhou e termina) e a utilização de pedras e de adicionar uma linha/coluna ao tabuleiro (pedra1, pedra2, lc1, lc2).

3. Estrutura dinâmica

Como referido no ponto 2, foi usado uma variável tabuleiro do tipo char**. Na figura abaixo encontra-se o código da inicialização do tabuleiro, neste caso foi usado uma estrutura dinâmica:

```
// inicializa o tabuleiro
char **inicializa_tab(pItab x) {
    int i, j;
    char **aux = NULL;

    aux = (char **) malloc(x->linha * sizeof(char *));
    if(aux == NULL){
        printf("\nErro a alocar memória!\n");
        return x->tabuleiro;
    }
    for (i=0; i<x->coluna; i++) {
        aux[i] = (char *) malloc(x->coluna * sizeof(char));
        if (aux[i] == NULL) {
            printf("\nErro a alocar memória!\n");
            return x->tabuleiro;
        }
    }
    for (i = 0; i < x->linha; i++) {
        for (j = 0; j < x->coluna; j++)
            aux[i][j] = '_';
    }
    return aux;
}
```

Esta função recebe a estrutura de dados Itab. Começo por declarar uma variável auxiliar (char **aux), para alocar memória para o tabuleiro. O primeiro malloc aloca espaço para uma coluna (são o número de linhas) e o segundo malloc aloca espaço para as restantes colunas, ou seja, em cada linha é necessário alocar espaço para uma coluna, esta é a função do segundo malloc.

Nas imagens a baixo encontra-se a lista ligada que guarda todas as jogadas realizadas:

```
pItab adiciona_lista(pItab lista, pItab x){

    char **tab = NULL;
    int i;
    int l,c;

    //Cria um novo tabuleiro
    tab = (char **) malloc(x->linha * sizeof(char *));
    if(tab == NULL){
        printf("\nErro a alocar memória!\n");
        return lista;
    }
    for (i=0; i<x->linha; i++) {
        tab[i] = (char *) malloc(x->coluna * sizeof(char));
        if (tab[i] == NULL) {
            printf("\nErro a alocar memória!\n");
            return lista;
        }
    }

    //Copia o tabuleiro para o novo tabuleiro
    for (l = 0; l < x->linha; l++) {
        for (c = 0; c < x->coluna; c++) {
            tab[l][c] = x->tabuleiro[l][c];
        }
    }

    pItab novo_tab = malloc(sizeof (Itab));
    if(novo_tab == NULL){
        fprintf(stderr, "Erro a alocar memória!");
        return lista;
    }

    *novo_tab = *x;
    novo_tab->tabuleiro = tab; //copia para a lista ligada o novo tabuleiro
```

```
if(lista == NULL){
    lista = novo_tab;
    novo_tab->prox = NULL;
} else {
    pItab aux = lista;
    while(aux->prox != NULL)
        aux = aux->prox;
    aux->prox = novo_tab;
    novo_tab->prox = NULL;
}

return lista;
}
```

Começo por criar um tabuleiro auxiliar para copiar o tabuleiro principal, depois é alocado espaço para a variável “novo_tab” com o tamanho da estrutura de dados, em seguida faço uma cópia da estrutura de dados para a lista ligada.

4. Manual de instruções

No início do jogo é apresentado um menu em que o utilizador tem que carregar na tecla 1 do teclado para iniciar o jogo.

```
-----Jogo do semaforo-----  
      1. Novo jogo.  
-----  
Opção: |
```

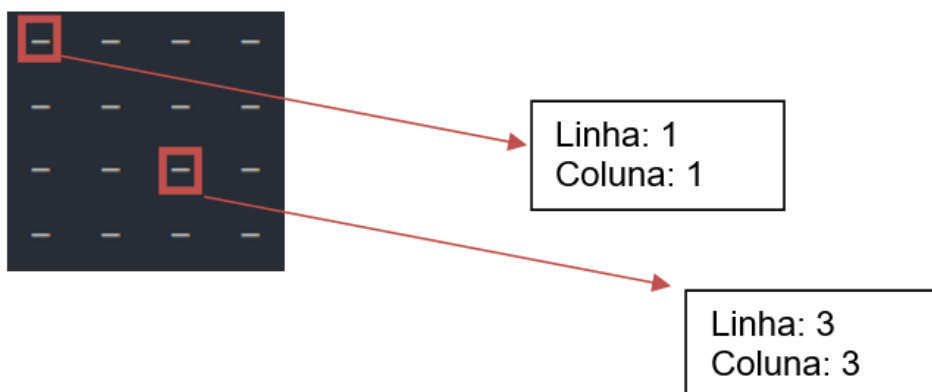
Após colocar o 1 a frente da opção aparece uma pergunta em que o utilizador tem que escolher se quer um jogo entre duas pessoas ou se quer um jogo contra o “computador”.

```
2 Jogadores ou 1 Jogador?  
Opção: |
```

Após seleccionar o número de jogadores é mostrado as dimensões do tabuleiro (3 – 5) e o tabuleiro em si. Logo após o utilizador tem que indicar onde pretende colocar uma peça verde (linha e coluna).

```
A tabela terá as seguintes dimensões -> Linhas: 4   Colunas: 4  
  
- - - -  
- - - -  
- - - -  
- - - -  
  
O jogador 1 irá colocar uma peça verde.  
  Indique a linha: 1  
  
  Indique a coluna: 1
```


Na figura a baixo encontra-se uma explicação num tabuleiro 4x4 o que o utilizador pode escrever no campo “Indique a linha” e no campo “Indique a coluna”.



De seguida na consola é apresentado o tabuleiro com a jogada realizada e um menu com todas as possibilidades que o jogador tem para jogar.

```
É a vez do jogador 2

----- Menu -----
0 que pretende fazer?
  1. Colocar uma peça verde.
  2. Mudar a cor.
  3. Adicionar uma pedra.
  4. Adiciona uma linha ou coluna.
  5. Ver as jogadas anteriores.
  0. Terminar o jogo.
-----
Opção: |
```

No campo “Opção” o jogador pode escolher entre colocar uma peça verde, mudar de cor, adicionar uma linha/coluna ver as jogadas anteriores e terminar o jogo.

Nas opções 1 (adiciona uma peça verde),2 (muda de cor para amarelo ou vermelho),3 (adiciona uma pedra) são apresentados os dois campos “Indique a linha”, “Indique a coluna” (que já foram explicados anteriormente) e logo em seguida é apresentado o tabuleiro com a jogada realizada e de novo é apresentado o menu.

Na opção 4 é apresentado o seguinte campo:

```
Opção: 4
Pretende aumentar a linha (1) ou a coluna (2): |
```

Caso o jogador decida adicionar uma linha deve colocar o algarismo “1” e caso pretenda adicionar uma coluna deve colocar o algarismo “2”.

Na opção 5 é apresentado o seguinte campo:

```
Opção: 5
Indique quantas jogadas quer ver: |
```

Assim o jogador deve indicar quantas jogadas quer ver. E será apresentado o jogador que realizou aquela jogada e o número da jogada e o tabuleiro com a respetiva jogada.

Exemplo:

```
Jogador = 2   N° Jogadas = 2
```

```
G  _  _  _
_  G  _  _
_  _  _  _
_  _  _  _
```

```
Jogador = 1   N° Jogadas = 3
```

```
G  _  _  _
_  G  _  _
_  _  _  _
_  _  _  _
_  _  _  _
```

```
É a vez do jogador 2
```

Na opção 0, o jogo simplesmente termina e é perdido todo o jogo efetuado até ao momento.

Quando um jogador ganha o jogo é apresentado o tabuleiro da última jogada e um campo para o utilizador escrever o nome de um ficheiro de texto que será gerado no final do jogo.

```
G  _ _ _ _
_  G _ _ _
_  _ G _ _
_  _ _ G _
_  _ _ _ G
_  _ _ _ G

O jogo vai criar um ficheiro de texto para que possa ver as jogadas efetuadas!
Por favor introduza um nome para o ficheiro: |
```

Nota: Se for um jogo só de 1 jogador o “computador” fará todas as opções que estão no menu, exceto a opção 5 e 0.

5. Conclusão

Com a realização deste trabalho, consegui aplicar em prática, os conhecimentos lecionados em aula e ultrapassar algumas das minhas dificuldades.

Sinto que este trabalho me ajudou a ser mais autónomo e a procurar soluções para as minhas dificuldades e para os problemas na realização deste trabalho.