Instituto Superior de Engenharia de Coimbra

Departamento de Engenharia Informática e de Sistemas
Licenciatura em Engenharia Informática
Programação 2021-2022

Relatório Trabalho Prático

Programação

Miguel Ferreira Neves – 2020146521

Índice

Organização do Programa

O programa tem o seu código fonte distribuído por vários ficheiros de forma a reduzir a quantidade de código e a melhorar a leitura do ficheiro principal *main.c*, recorrendo a utilização de *header files* para gerir as dependências entre os vários ficheiros.

Tendo assim o código organizado por vários ficheiros:

- ***matrix***, contém funções onde engloba a interação de uma matriz dinâmica.
- ***board***, contém funções com a utilização das listas ligadas.
- ***files***, com funções para a interagir com ficheiros binários e de texto.
- ***utils***, ficheiro com um conjunto de funções úteis, como, geração de um número aleatório ou reagrupar um trecho de código do ficheiro principal *main.c* de forma a o deixar mais limpo possível.

Estruturas

O programa recorre a duas listas ligadas diferentes, uma para armazenamento dos mini tabuleiros outra para armazenar as jogadas feitas ao longo do jogo.

- ***miniBoard***, esta estrutura possui:
  - Um *id*, uma variável com um número inteiro variando entre 1 e 9.
  - O *miniBoard* em questão, que não é mais nada que uma matriz alocada dinamicamente.
  - E por fim um ponteiro (*next*) para a próxima estrutura ou para *NULL* caso for a última estrutura da lista ligada.

```
typedef struct miniBoard miniBoard, *pMiniBoard;
struct miniBoard{
    int id;
    char **miniBoard;
    pMiniBoard next;
};
```

A forma de como o tabuleiro do jogo foi implementado não é de todo a maneira mais simples e otimizada. Um *array* de estruturas por exemplo, utilizar-se-ia menos memoria visto que, o *id* e o ponteiro *next* deixariam de existir. Neste programa o impacto seria mínimo, mas em programas com maior dimensão esse impacto poderia ser mais significativo.

- ***Plays***, esta estrutura contém:
  - *nPlays*, variável inteira que é incrementada ao alocar uma nova estrutura, numerado o número de jogadas.
  - *player*, variável inteira que varia entre 1 e 2 consoante o jogador que fez a jogada.
  - *boardID*, variável inteira é definida com o id do mini tabuleiro onde a jogada aconteceu, variando entre 1 e 9.
  - *linePlayed*, variável inteira que contem a linha onde o respetivo jogador jogou, neste caso varia entre 0 e 2.
  - *columnPlayed*, variável inteira que contem a coluna onde o respetivo jogador jogou, varia neste caso entre 0 e 2.
  - Por fim e não menos importante, um ponteiro *next* que aponta para a próxima estrutura ou para *NULL* se for a última estrutura da lista ligada.

```
typedef struct plays plays, *pPlays;
struct plays{
    int nPlays;
    int player;
    int boardID;
    int linePlayed;
    int columnPlayed;
    pPlays next;
};
```

Opções de Implementação e Regras do Jogo

Antes de iniciar um novo jogo, o programa verifica se existe um jogo por continuar e questiona o utilizador se pretende continuar esse mesmo jogo.

O jogo começa num mini tabuleiro aleatório entre 1 e 9.

Enquanto um mini tabuleiro não tiver totalmente preenchido será sempre possível jogar nele, mesmo que já haja um vencedor, que será o vencedor desse mini tabuleiro até ao final do jogo.

Neste programa tive o "cuidado" de implementar o código em inglês (para um ponto de vista mais profissional), exceto tudo o que é comentários e introdução de dados na consola (*printf*), como também a verificação dos dados introduzidos pelo utilizador nas medidas possíveis da linguagem *c*.

Conclusão

Neste trabalho prático tive em conta os requisitos do enunciado como a implementação de código em “inglês” como já disse posteriormente e também levei em consideração a outros pequenos pormenores como quantidade de código contido no meu ficheiro *main.c* ou a validação de parâmetros introduzidos pelo utilizador, colocando a implementação de código a um nível tanto escolar como profissional. O trabalho prático foi realizado e desenvolvido nos sistemas operativos da Microsoft Windows 10 e Windows 11, recorrendo ao IDE *Visual Studio Code* e compilado com *ming64*.