

Programação I 2019/2020

Departamento de Informática, Universidade de Évora

Trabalho prático
dezembro de 2019 (v2.0)

— 2048 —

1 Introdução

2048 é um jogo de quebra-cabeça para um jogador. O objetivo do jogo é deslizar peças numeradas numa grelha de modo a combiná-las até criar uma peça com o número 2048. É um jogo viciante e muitas pessoas passam muitas horas a jogar este jogo (em vez de trabalhar, estudar ou mesmo dormir!)

As peças podem ser movimentadas em 4 direções: **Baixo**, **Cima**, **Direita** e **Esquerda**. e são combinadas quando são adjacentes e têm o mesmo número. Neste caso, são substituídas por uma nova peça cujo valor é a soma das duas peças iniciais.

2 Descrição do trabalho

O trabalho consiste em desenvolver aplicações para jogar o *2048* em dois modos distintos:

- No **modo iterativo** a grelha inicial é preenchida em duas posições (aleatorias), com o número 2 ou 4. Em cada jogada,
 1. o jogador escolhe um sentido (**B**, **C**, **D** e **E**);
 2. a grelha é atualizada de acordo com as regras do jogo;
 3. é colocado um novo número 2 ou 4 numa posição vazia.

O jogo termina quando o utilizador escolhe **F** (**Fim**) ou quando não for possível combinar mais peças; nessa altura é apresentado o número total de peças combinadas durante o jogo e a contagem do nº de peças com cada número ainda no tabuleiro (por ordem crescente).

- No **modo automático**, a grelha inicial (totalmente preenchida) e as jogadas são lidas de um ficheiro de texto e o programa apresenta o número total de peças combinadas durante o jogo e a contagem após a última jogada.

Por exemplo, para o tabuleiro de dimensão 4 abaixo, temos a seguinte sequência de configurações quando as jogadas são (por ordem): **B**, **D** e **B**.

2	2	4	2	⇒	-	-	-	-	⇒	-	-	-	-	⇒	-	-	-	-
4	2	2	4	⇒	2	2	4	2	⇒	-	4	4	2	⇒	-	-	4	2
2	2	2	2	⇒	4	4	2	4	⇒	-	8	2	4	⇒	-	4	2	4
2	4	2	2	⇒	4	4	4	4	⇒	-	-	8	8	⇒	-	8	8	8

3 Desenvolvimento

O jogo deve ser implementado na linguagem **C** e ser acompanhado por um relatório **PDF**.

Para o desenvolvimento do trabalho assuma que o **tamanho máximo** do tabuleiro é 10*10 quadrados. Deverá **obrigatoriamente** implementar as seguintes funções:

```
int baixo(int grelha[][], int sz)
```

Esta função atualiza a grelha ao ser escolhido o sentido **B** (Baixo). A função devolve o nº de peças combinadas.

```
int cima(int grelha[][], int sz)
```

Esta função atualiza a grelha ao ser escolhido o sentido **C** (Cima). A função devolve o nº de peças combinadas.

```
int direita(int grelha[][], int sz)
```

Esta função atualiza a grelha ao ser escolhido o sentido **D** (Direita). A função devolve o nº de peças combinadas.

```
int esquerda(int grelha[][], int sz)
```

Esta função atualiza a grelha ao ser escolhido o sentido **E** (Esquerda). A função devolve o nº de peças combinadas.

```
int jogada(int grelha[][], int sz, char sentido)
```

Esta função executa uma jogada, devolvendo o número de peças combinadas. Deve utilizar as funções e procedimentos anteriores.

```
void mostrar(int grelha[][], int sz)
```

Este procedimento mostra no ecrã a configuração atual da grelha.

Deverá desenvolver um **programa distinto para cada um dos modos de funcionamento** mas re-utilizando as funções comuns a ambos os modos de funcionamento. Para tal sugere-se a utilização dos seguintes ficheiros:

- 2048.h, com os protótipos das funções comuns a ambos os programas;
- 2048.c, com a implementação das referidas funções;
- iterativo.c, com o jogo para o modo iterativo;
- automatico.c, com o jogo para o modo automático.

O comando

```
gcc -c 2048.c
```

compila o ficheiro 2048.c e gera o ficheiro objeto 2048.o.

O comando

```
gcc -o iterativo 2048.o iterativo.o
```

liga os ficheiros objeto (2048.o e iterativo.o) e cria o executável iterativo.

3.1 Modo iterativo

Este modo implementa o jogo "oficial". Funciona da seguinte forma:

1. é solicitado ao utilizador o tamanho da grelha N;

2. é apresentada no ecrã uma grelha de tamanho $N \times N$, com os algarismos 2 ou 4 em 2 posições (aleatórias);
3. iterativamente, o jogador escolhe um sentido, sendo apresentada no ecrã a grelha atualizada, mais um novo algarismo (2 ou 4) numa posição livre. O jogo termina quando não é possível fazer mais nenhuma operação ou quando o utilizador escolhe F.
4. nessa altura é apresentado o número total de peças combinadas e o nº de peças de cada número na grelha final (ver modo automático).

Para gerar os algarismos e a posição onde os mesmos são colocados deve utilizar a função `int rand()` da biblioteca `stdlib.h`. Esta função gera um número inteiro aleatório entre 0 e `RAND_MAX`. O operador `%` (resto da divisão inteira) é utilizado em conjunto com a função `rand()` para obter valores em intervalos diferentes.

3.2 Modo automático

No modo automático a grelha inicial está totalmente preenchida, sendo toda a informação lida de um ficheiro. A **informação de entrada** contém:

- uma linha com o tamanho da grelha N ;
- N linhas com N algarismos cada uma;
- uma linha com as K jogadas

A **resposta**, a apresentar no ecrã, contém, igualmente, o nº total de peças combinadas e o nº de peças restantes (de cada número).

Por exemplo, para o ficheiro com a seguinte informação:

```
4
2 2 4 2
4 2 2 4
2 2 2 2
2 4 2 2
B D B E C
```

deverá ser apresentado no ecrã

```
pecas combinadas: 11
contagem: 0  2  2  1
```

4 Relatório

A organização do relatório e a qualidade do texto também são avaliadas. O texto do relatório deve ser escrito de forma impessoal, isto é, usando voz passiva, focando nos aspectos técnicos (e não nos autores). Exemplos:

- Certo: Foi encontrado um erro...
- Errado: Encontrámos um erro...

- **Certo:** Nos testes efectuados verificou-se que o programa...
- **Errado:** Reparámos que o programa não funciona correctamente...
- **Certo:** A função xpto foi usada para...
- **Errado:** Usámos a função xpto para...

Deve fazer a descrição da solução proposta e incluir toda a informação relativa ao seu desenvolvimento, incluindo as variáveis utilizadas (e respetivos tipos) e a lista de funções implementadas especificando os argumentos recebidos, o valor devolvido e uma descrição detalhada da funcionalidade implementada (não incluir código). Para ter uma ideia do que se pretende, veja as *man pages* em Linux usando o comando `man`; experimente, por exemplo, `man puts`.

5 Prazos e Entrega

O trabalho deve ser desenvolvido em grupos de **2 alunos** e submetido no moodle até ao final do dia **12 de janeiro de 2019**. A **discussão** do trabalho será feita na semana seguinte.

A submissão deve incluir o(s) ficheiro(s) de código bem como o relatório. Para tal deverá comprimir todos os ficheiros para um de nome definido pelos números dos alunos; por exemplo, para o grupo com alunos nº 2345 e nº 6666, o ficheiro submetido deverá ter o nome `2345_6666.zip`.

O trabalho é **obrigatório para todos os alunos** inscritos à disciplina (mesmo que tenham feito um trabalho o ano passado).

Bom trabalho!