

# Programação I 2018/2019

Departamento de Informática, Universidade de Évora

## Trabalho prático

dezembro de 2018

### – Color Squares –

## 1 Introdução

Este é um jogo bem conhecido: um tabuleiro (vertical) é preenchido com quadrados de cores diferentes e o jogador pode remover grupos com a mesma cor; um grupo consiste num conjunto de quadrados com lados partilhados. Após cada movimento, o tabuleiro é atualizado de acordo com as seguintes regras:

1. *gravidade*: Os quadrados acima da área vazia caem devido à gravidade;
2. *coluna*: Quando toda a coluna está vazia, ela colapsa movendo os blocos da direita para a esquerda para fechar a separação.

## 2 Descrição do trabalho

O trabalho consiste em desenvolver uma aplicação para jogar *Color Squares*. Existem 2 modos de jogo: o modo interativo e o modo automático:

- No **modo interativo** o tabuleiro inicial é gerado definindo aleatoriamente uma de quatro cores possíveis (1, 2, 3 e 4) para cada posição do tabuleiro. Em cada jogada, o jogador escolhe um dos quadrados do tabuleiro, sendo este atualizado de acordo com as regras do jogo até que o tabuleiro fique vazio; nessa altura é apresentada a pontuação final.
- No **modo automático**, o tabuleiro inicial e as jogadas são lidas de um ficheiro de texto e o programa deve apresentar a pontuação final.

Por exemplo, para um tabuleiro de dimensão 4, temos a seguinte sequência de configurações quando o jogador faz as jogadas:

- escolhe o 4 na posição (2,2). A posição (0,0) é o canto inferior esquerdo.
- escolhe o 3 na posição (2,1)

1	2	3	2		1	-	-	2		1	-	2	-
1	3	4	2	→	1	2	-	2	→	1	-	2	-
2	4	4	1		2	3	3	1		2	2	1	-
1	2	3	2		1	2	3	2		1	2	2	-

O jogador ganha B ponto quando remove um grupo de um único quadrado; um grupo de dois quadrados pontua R pontos, enquanto um grupo de três pontua 6 e assim por diante. A fórmula seguinte indica o n° de pontos de uma jogada:

$$pontos = \frac{num\_quadrados * (num\_quadrados + 1)}{2}$$

### 3 Desenvolvimento

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

Para o desenvolvimento do trabalho deverá **obrigatoriamente** implementar as seguintes funções:

`int marcar(int tabuleiro[][], int sz, int x, int y)`

Esta função marca no tabuleiro todos os quadrados que fazem parte do grupo do quadrado (x,y). A posição (0,0) é o canto inferior esquerdo. A função devolve o n° de quadrados do grupo.

`int pontuacao(int num_quadrados)`

Esta função calcula a pontuação de uma jogada.

`void gravidade(int tabuleiro[][], int sz)`

Este procedimento atualiza o tabuleiro de acordo com a regra da *gravidade*: a área vazia é preenchida pelos quadrados acima da mesma.

`void coluna(int tabuleiro[][], int sz)`

Este procedimento atualiza o tabuleiro de acordo com a regra da *coluna* vazia: quando toda a coluna está vazia, ela colapsa movendo os quadrados da direita para a esquerda para fechar a separação.

`int jogada(int tabuleiro[][], int sz, int x, int y)`

Esta função executa uma jogada, devolvendo a pontuação da mesma. Deve utilizar as funções e procedimentos anteriores.

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

Este procedimento mostra no ecrã a configuração atual do tabuleiro.

Deverá desenvolver um programa distinto para cada um dos modos de funcionamento mas re-utilizando (pelo menos) as funções atrás descritas. Para tal sugere-se utilização dos seguintes ficheiros:

- `colorSquares.h`, com os protótipos das funções comuns a ambos os programas;
- `colorSquares.c`, com a implementação das referidas funções;
- `csIter.c`, com o jogo para o modo iterativo;
- `csAuto.c`, com o jogo para o modo automático.

O comando

```
gcc -c colorSquare.c
```

compila o ficheiro `colorSquare.c` e gera o ficheiro objeto `colorSquares.o`.

O comando

```
gcc -o csIter colorSquares.o csIter.o
```

liga os ficheiros objetos e cria o executável `csIter`.

### 3.1 Modo iterativo

Este modo funciona da seguinte forma:

1. é solicitado ao utilizador o tamanho do tabuleiro  $N$ ;
2. é gerado um tabuleiro de tamanho  $N*N$  e apresentado no ecrã;
3. iterativamente, e até o tabuleiro estar vazio, o jogador escolhe um quadrado, sendo apresentado no ecrã o novo tabuleiro resultante da remoção do grupo a que esse quadrado pertence.

Para criar o tabuleiro inicial deve utilizar a função `int rand()` da biblioteca `stdlib.h`; esta função gera um número inteiro aleatório entre 0 e `RAND_MAX`. É prática comum usar o operador `%` em conjunto com `rand()` para obter valores num intervalo diferente.

### 3.2 Modo automático

No modo automático o objetivo é, dado um tabuleiro inicial e sequência de movimentos, mostrar a pontuação final do jogo. A **informação de entrada** contém:

- uma linha com o tamanho do tabuleiro  $N$ ;
- $N$  linhas de  $N$  algarismos cada um representando a cor do quadrado respetivo;
- uma linha com a contagem de movimentos  $K$  descritos abaixo;
- uma linha com  $K$  pares  $(X_i Y_i)$  separados por vírgula. Estas são as coordenadas - coordenadas dos movimentos (onde  $Y=0$  é a linha inferior e  $X=0$  é a coluna mais à esquerda).

A **resposta**, a apresentar no ecrã, deve conter um único valor – a pontuação final.

Por exemplo, o valor apresentado no ecrã deve ser 54 para o ficheiro com a seguinte informação:

```
5
14111
23334
43212
33323
21232
12
0 4, 3 0, 0 1, 2 2, 1 1, 4 2, 3 2, 0 1, 3 2, 3 1, 3 0, 2 0
```

## 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 não funciona correctamente se...
- **Errado:** Reparámos que o programa não funciona correctamente se...
- **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. 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 **3 alunos** e submetido no moodle até ao final do dia **20 de janeiro de 2019**. A **discussão** do trabalho será feita na semana seguinte.

No dia **4 de janeiro às 16:00** haverá uma sessão de apresentação e esclarecimento de dúvidas sobre o trabalho.

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 1111, nº 2345 e nº 6666, o ficheiro submetido deverá ter o nome `1111_2345_6666.zip`.

*Bom trabalho!*