



UNIVERSIDADE
DE ÉVORA

Programação I - Color Squares - Trabalho prático - Relatório

Trabalho realizado por:

Rui Roque nº42720

Tomás Dias nº42784

Introdução

Foi proposto na cadeira de **Programação I** o desenvolvimento de uma aplicação, em linguagem **C**, para jogar *Color Squares*.

A descrição deste jogo encontra-se no enunciado cedido para a realização deste relatório, destacando-se o funcionamento do jogo em si e as duas regras básicas do mesmo: a regra *gravidade* e a regra *coluna*. Com esta informação, o trabalho teria de ser criado com base em dois modos de jogo: o modo **interativo** e o modo **automático**. Como se sucede com a descrição do jogo, a descrição destes modos também se encontra no enunciado cedido. Também se encontra descrito o funcionamento da pontuação da aplicação.

Com estas informações estudadas e acumuladas, prosseguiu-se para o desenvolvimento da dita aplicação.

Desenvolvimento

Antes de qualquer procedimento prático, foi necessário perceber como o jogo iria ser criado. Assim, definiu-se que:

- A aplicação seria constituída por uma matriz quadrada (nxn) que corresponde ao tabuleiro de jogo, sendo o tamanho máximo do tabuleiro de 20x20 quadrados.
- Este tabuleiro seria formado por quatro cores sendo estas cores apresentadas pelos números: 1, 2, 3 e 4.

Para a criação do jogo, foi obrigatório terem sido implementadas as seguintes **funções**:

- `int marcar();`
- `int pontuação();`
- `void gravidade();`
- `void coluna();`
- `int jogada();`
- `void mostrar();`

Com isto, procurou-se a funcionalidade destas.

Funções

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

Esta função recebeu como **argumentos**:

- **int tabuleiro[][]**: variável que representa a matriz/tabuleiro do jogo com a dita dimensão;
- **int sz**: variável que representa o tamanho da matriz quadrada, isto é, indica o número de linhas e colunas da mesma;
- **int x**: variável que representa o valor da coluna;
- **int y**: variável que representa o valor da linha.

Como funcionalidade, esta marca no tabuleiro todos os quadrados que fazem parte do grupo do quadrado (x,y), sendo a posição (0,0) o canto inferior esquerdo, devolvendo o número de quadrados do grupo.

Função **int pontuacao(int num_quadrados)**

Esta função recebeu como **argumentos**:

- **int num_quadrados**: variável que representa o número de um grupo de quadrados com o mesmo valor (1,2,3 ou 4).

Como funcionalidade, esta recorre à equação dos pontos:

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

devolvendo o número de pontos ganhos pelo jogador em cada jogada.

Função: **void gravidade(int tabuleiro[][], int sz)**

Esta função tem como **argumentos**:

- **int tabuleiro[][]**: variável que representa a matriz/tabuleiro do jogo com a dita dimensão;
- **int sz**: variável que representa o tamanho da matriz quadrada, isto é, indica o número de linhas e colunas da mesma.

Também foi utilizada nesta função a variável **int posv**. Esta variável corresponde à linha que contém um 0.

Tem como funcionalidade atualizar o tabuleiro de acordo com a regra da *gravidade*: a área vazia é preenchida pelos quadrados acima da mesma. Sendo uma função do tipo void, esta não devolve nenhum valor.

Função: **void coluna(int tabuleiro[][], int sz)**

Esta função tem como **argumentos**:

- **int tabuleiro[][]**: variável que representa a matriz/tabuleiro do jogo com a dita dimensão;
- **int sz**: variável que representa o tamanho da matriz quadrada, isto é, indica o número de linhas e colunas da mesma.

Também foi utilizada nesta função a variável **int colv**. Esta variável corresponde a uma coluna apenas constituída por 0's.

Tem como funcionalidade atualizar o tabuleiro de acordo com a regra da *coluna*: quando toda a coluna está vazia, ela colapsa movendo os quadrados da direita para a esquerda para fechar a separação. Sendo uma função do tipo void, esta não devolve nenhum valor.

Função: int jogada(int tabuleiro[][], int sz, int x, int y)

Esta função tem como **argumentos**:

- **int tabuleiro[][]**: variável que representa a matriz/tabuleiro do jogo com a dita dimensão;
- **int sz**: variável que representa o tamanho da matriz quadrada, isto é, indica o número de linhas e colunas da mesma;
- **int x**: variável que representa o valor da coluna;
- **int y**: variável que representa o valor da linha.

Esta função executa uma jogada, devolvendo a pontuação da mesma, utilizando as funções: **marcar()**; **pontuação()**.

Função: void mostrar(int tabuleiro[][], int sz)

Esta função tem como **argumentos**:

- **int tabuleiro[][]**: variável que representa a matriz/tabuleiro do jogo com a dita dimensão;
- **int sz**: variável que representa o tamanho da matriz quadrada, isto é, indica o número de linhas e colunas da mesma.

Esta função devolve a configuração atual do tabuleiro, mostrando-a no ecrã de jogo.

Modo interativo

Neste modo, utilizando a função **int main()**, pede-se um input ao jogador ao qual corresponde a variável **int n**. Esta variável representa o número de linhas (ou colunas) da matriz quadrada

De seguida, gera-se um tabuleiro com as dimensões especificadas pelo jogador. Para que este procedimento ocorra, é necessário a criação da função **int rand()**. Esta função não recebe argumento e gera um número inteiro aleatório entre 0 e **RAND_MAX**.

Mas para evitar que os valores gerados se repitam e que a função **int rand()** seja inicializada de modo a produzir um valor aleatório na faixa determinada, é preciso a utilização da função **srand()**. Esta função tem um argumento do tipo **int unsigned**.

Estas funções encontram-se na biblioteca **stdlib.h**.

Após isto, implementa-se as restantes funções, de modo a que o jogo seja executável de acordo com a mecânica e regras do mesmo.

Erros neste modo

Alguns erros ocorreram no desenvolvimento deste modo, estes causados pela possível deficiente implementação das funções que constituem a aplicação, na **função int main()**. Verificou-se que quando o programa era executado, aparecia a informação para que o utilizador introduzisse o input e que de seguida, o programa gerava uma matriz corretamente com os valores aleatórios. Mas após isso, apesar de aparecer a informação para que o utilizador colocasse as coordenadas para marcar um número da matriz, o programa não executava as restantes funções, impossibilitando assim que a aplicação não efetuasse o pretendido, isto é, não era possível ao utilizador jogar.

Modo automático

Com os problemas ocorridos no desenvolvimento do modo interativo, não foi possível a conclusão deste modo de jogo, apenas conseguiu-se criar os 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.

