



Escola de Ciências e Tecnologias
Departamento de Informática
Licenciatura em Engenharia Informática

Programação 1
(1º ano, 1º semestre, 2019/2020)
Docente: Prof.^a Doutora Teresa Gonçalves

Relatório do trabalho prático – 2048 –

Discentes:
Alexandre Costa nº15856

12 de janeiro de 2020
Évora

Índice:

Introdução.....	p.2
Estrutura do Trabalho.....	p.3
Funções.....	p.4
Aspetos finais.....	p.18

Introdução:

O objetivo deste trabalho é fazer um programa de uma versão do jogo 2048 com o conhecimento adquirido nas aulas.

O jogo foi realizado em linguagem C como recurso a variadas funções sendo estas utilizadas de várias maneiras para os dois modos do jogo, modo interativo e modo automático.

O modo interativo consiste numa grelha que o utilizador pretende, em que aleatoriamente dispõe dois números(2 ou 4) na grelha, de seguida pede uma jogada ao utilizador (baixo, cima, direita ou esquerda) depois executa-a e na grelha e apresenta o resultado da jogada ao utilizador continuamente até o utilizador ficar sem jogadas possíveis ou decidir que não quer continuar o jogo.

O modo automático recebe um ficheiro com informações sobre o tamanho da grelha, a grelha em si e as suas jogadas que no final devolve o número total de peças combinadas depois da última jogada do ficheiro.

Estrutura do trabalho:

- O trabalho prático tem dois modos de utilização, o modo iterativo e o modo automático, onde ambos os modos de jogo fazem uso da biblioteca “2048.h” e do ficheiro “2048.c”. Utilizando as funções neles criadas para o funcionamento dos dois modos de jogo.

- Modo iterativo: apresenta um breve tutorial do modo de jogo, pedindo de seguida o tamanho da grelha pretendido para que o jogo seja jogado, fazendo uso da função `InstrucoesDoJogo_TamanhoDaGrelha`. Posteriormente, mostra a grelha com dois números gerados aleatoriamente em posições aleatórias e solicita ao utilizador uma jogada dando a possibilidade de terminar o jogo.

A cada jogada efetuada pelo jogador, o programa verifica se a jogada é possível, se existem espaços disponíveis para gerar novo número aleatório e mostra a grelha atualizada, usando as funções `Verifica_jogada_possivel`, `verifica_espacos_vazios` e `Mostrar`. O jogo termina quando a função `Verifica_jogada_possivel` indica que não é possível efetuar um movimento ou quando o utilizador escolhe a opção “[F]im”. Nesse momento, é executada a função `Fim_do_Jogo` para que seja apresentado o número de combinações e a contagem de cada tipo de peça.

- Modo automático: para que o modo de jogo seja jogado é necessário que o utilizador insira um ficheiro corretamente formatado. Posteriormente o programa lê o primeiro caracter do ficheiro para que seja criada uma grelha com a dimensão correspondente. Faz uso de dois ciclos `for` para que essa mesma grelha seja preenchida com os valores do ficheiro e de um ciclo `while` que irá ser executado até o programa acabar de ler o ficheiro, a cada caracter que é lido do ciclo `while` a condição `switch` lê o caracter e chama a função correspondente ao movimento. Quando o programa termina de ler o ficheiro é executada a função `Fim_do_Jogo` para que seja apresentado o número de combinações e a contagem de cada tipo de peça.

Funções:

- InstrucoesDoJogo_TamanhoDaGrelha:

Objetivo:

- Apresentar as instruções do jogo ao utilizador e pede o tamanho da grelha ao utilizador.

Argumentos:

- Não tem.

Valor de Retorno:

- A função devolve a variável “sz” que corresponde ao tamanho da grelha aceitando apenas valores entre 2 e 10.

Variáveis Auxiliares:

- sz, variável do tipo inteiro que guarda o tamanho da grelha.

Procedimento:

- Inicializa a variável “sz” com o valor de 0;
- Informa ao utilizador as instruções do modo iterativo;
- Solicita que seja inserido pelo o utilizador o tamanho da grelha usando um ciclo while para garantir que o tamanho da grelha seja entre 2 e 10;
- Devolve a variável “sz”.

- Inicia_Grelha

Objetivo:

- Atribuir o valor “0” a todas as posições da grelha e de seguida, utilizando a função “int rand()” da biblioteca “<stdlib.h>”, seleccionar de uma maneira aleatória duas posições distintas e colocar aleatoriamente os valores 2 ou 4 nessas mesmas posições.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de Retorno:

- Não tem.

Variáveis Auxiliares:

- colunaescolhida, inteiro. Guarda um valor aleatório entre 0 e “sz” correspondendo à coluna da grelha;
- linhaescolhida, inteiro. Guarda um valor aleatório entre 0 e “sz” correspondendo à linha da grelha;
- escolha, inteiro. Aleatoriamente guarda o valor 2 ou 4.

Funções Auxiliares:

- int rand().

Procedimento:

- Utiliza um ciclo for para preencher cada posição da grelha com o valor 0;
- Foi usada a função rand() para aleatoriamente escolher o valor 2 ou 4 e guarda-lo na variável escolha;
- Foi utilizado um novo ciclo for para colocar o valor guardado na variável escolha em duas posições aleatórias da grelha. Nesse mesmo ciclo for as variáveis colunaescolhida e linhaescolhida também fizeram uso da função rand() para que a posição da grelha fosse aleatória.

- **Mostrar:**

Objetivo:

- Mostrar no ecrã a configuração da grelha atual substituindo os valores de “0” pelo caracter “-“.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de Retorno:

- Não tem.

Procedimento:

- Foi fornecida à função Mostrar as variáveis “sz” e “grelha” do tipo inteiro;
- A função faz uso de uma condição if dentro de dois ciclos for para que no percorrer da grelha quando encontra um valor “0” substitua pelo caracter “-“, caso o valor seja diferente de “0” mostra esse mesmo valor.

- **verifica_espacos_vazios:**

Objetivo:

- Verificar quantos espaços vazios existem em cada posição da grelha.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de Retorno:

- espacosvazios, inteiro.

Variáveis Auxiliares:

- espacosvazios, inteiro.

Procedimento:

- Inicializada a variável “espacosvazios” atribuindo-lhe o valor “0”;
- Utilizado dois ciclos for para percorrer as colunas e linhas da grelha e quando encontra o valor “0” incrementa um valor à variável “espacosvazios”;
- Devolve à variável o valor da variável “espacosvazios”.

- **gera_novo_numero**

Objetivo:

- Gerar um número aleatório (2 ou 4) e inseri-lo numa posição aleatória da grelha desde que essa posição tenha o valor “0”.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de Retorno:

- Não tem.

Variáveis Auxiliares:

- numeroaleatorio, inteiro;
- numeropossivel, inteiro;
- l, inteiro;
- c, inteiro.

Funções Auxiliares:

- int rand().

Procedimento:

- Inicializadas as variáveis “numeroaleatorio”, “numeropossivel”, “l” e “c”;
- A variável “numeroaleatorio” faz uso da função rand() para receber o valor 2 ou 4 e a variável “numeropossivel” recebe o valor “0”;
- Utilizado o ciclo while enquanto a variável “numeropossivel” for igual a “0”, gera aleatoriamente um valor para as variáveis “l” (linha) e “c” (coluna) com o uso da função rand(), utilizando a condição if dentro do ciclo while verifica se essa posição da grelha tem o valor “0”. Caso a condição seja verdadeira substitui o valor “0” pela variável “numeroaleatorio” e atribui à variável “numeropossivel” o valor “1” parando o ciclo while; caso a condição seja falsa volta a gerar um novo valor para as variáveis “l” e “c”.

- **jogada**

Objetivo:

- A função jogada foi usada para executar o movimento que o utilizador pretender que poderá ser baixo, cima, direita, esquerda ou fim e de seguida gerar um novo numero e mostrar a grelha atualizada.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro;
- sentido, string.

Valor de Retorno:

- pecascombinadas, inteiro.

Variáveis Auxiliares:

- pecascombinadas, inteiro;
- existeespacovazio, inteiro.

Funções Auxiliares:

- verifica_espacos_vazios;
- baixo;
- cima;
- direita;
- esquerda;
- Mostrar;
- gerar_novo_numero.

Procedimento:

- Foi chamada a função verifica_espacos_vazios e atribuída à variável “existeespacovazio”.
- Utilizado uma condição switch que recebe a variável “sentido” correspondendo ao movimento que o utilizador pede. Caso o sentido seja diferente de F (Fim), executa a jogada pedida pelo utilizador e devolve para a variável “pecascombinadas” o número de peças combinadas na jogada executada e de seguida faz uso de uma condição if para

verificar se a variável “existeespacovazio” é diferente de “0”. Caso seja, é gerado um novo número aleatório na grelha utilizando a função gera_novo_numero.

Por fim é chamada a função Mostrar para que seja apresentada a grelha atualizada.

- Caso na condição switch o utilizador insira F (Fim), não é chamada nenhuma função auxiliar terminando assim a condição switch.
- A função jogada devolve o numero de peças combinadas, pela jogada solicitada pelo utilizador, na variável “pecascombinadas”.

- Verifica_jogada_possivel

Objetivo:

- A função tem como objetivo criar uma grelha auxiliar igual à grelha atual e verificar se é possível executar algum movimento.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de retorno:

- Devolve o valor 0 ou 1.

Variáveis auxiliares:

- Copiagrelha[sz][sz], inteiro.

Funções auxiliares:

- verifica_espacos_vazios;
- baixo;
- cima;
- direita;
- esquerda.

Procedimento:

- Inicializado um ciclo for para copiar a grelha atual para a variável “copiagrelha”.
- Criada a condição if para caso as funções verifica_espacos_vazios, baixo, cima, direita e esquerda devolvam 0, a condição if faz com que a função devolva 0, ou seja, que não foi possível efetuar nenhum movimento. Caso contrário, a função devolve o valor 1.

- Fim_do_jogo

Objetivo:

- A função tem como objetivo mostrar o fim do jogo. Mostra o numero de peças combinadas e a contagem do numero de peças de cada tipo no tabuleiro.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro;
- mombinadas, inteiro.

Valor de retorno:

- Não tem.

Variáveis auxiliares:

- maiornumero, inteiro;
- num, inteiro;
- contarepetidas, inteiro;

Funções auxiliares:

- pow(), disponível na biblioteca <math.h>.

Procedimento:

- Utilizados 2 ciclos for que percorrer a grelha e encontra o maior número guardando, esse mesmo, na variável “maiornumero”.
- Faz uso do argumento “combinadas” para mostrar ao utilizador quantas peças foram combinadas no percorrer do jogo.
- Utiliza um ciclo while que termina quando a função pow() devolve um valor maior ou igual à variavel “maiornumero”. Dentro do ciclo while existem dois ciclos for para percorrer a grelha e contabilizar o numero de peças iguais à condição do ciclo while. Faz uso de uma condição if para guardar o numero de peças repetidas na variável “contarepetidos”.

- No percorrer do ciclo while a função mostra ao utilizador o numero de peças repetidas de cada peça.

- baixo

Objetivo:

- A função baixo tem como objetivo mover todas as peças do tabuleiro para baixo.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de retorno:

- pecascombinadas, inteiro.

Variaveis auxiliares:

- pecascombinadas, inteiro;
- c, inteiro;
- l, inteiro;
- aux, inteiro.

Procedimento:

- A função inicia com dois ciclos for para percorrer a grelha.
- Dentro dos ciclos for existe uma condição if que verifica se o valor da posição da grelha é diferente ou igual a 0.
- Caso seja diferente, inicializa um ciclo while que verifica se o valor da posição na linha a baixo à correspondente é igual a zero ou igual à posição atual, iniciando uma condição if:
 - Caso o valor da posição da linha a baixo à correspondente seja igual à posição atual, o valor da posição atual é somado ao valor da posição a baixo dessa mesma posição e a posição atual toma o valor de 0.
 - Caso o valor da posição da linha a baixo à correspondente seja igual a 0, o valor da posição da linha atual será transportado para a posição da linha a baixo à correspondente e a posição atual toma o valor de 0.
- Caso o valor da posição da grelha seja igual a 0 os dois ciclos for são incrementados.
- Por fim, a função retorna o número de peças combinadas na jogada.

- cima

Objetivo:

- A função cima tem como objetivo mover todas as peças do tabuleiro para cima.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de retorno:

- pecascombinadas, inteiro.

Variaveis auxiliares:

- pecascombinadas, inteiro;
- c, inteiro;
- l, inteiro;
- aux, inteiro.

Procedimento:

- A função inicia com dois ciclos for para percorrer a grelha.
- Dentro dos ciclos for existe uma condição if que verifica se o valor da posição da grelha é diferente ou igual a 0.
- Caso seja diferente, inicializa um ciclo while que verifica se o valor da posição na linha a cima à correspondente é igual a zero ou igual à posição atual, iniciando uma condição if:
 - Caso o valor da posição da linha a cima à correspondente seja igual à posição atual, o valor da posição atual é somado ao valor da posição a cima dessa mesma posição e a posição atual toma o valor de 0.
 - Caso o valor da posição da linha a cima à correspondente seja igual a 0, o valor da posição da linha atual será transportado para a posição da linha a cima à correspondente e a posição atual toma o valor de 0.
- Caso o valor da posição da grelha seja igual a 0 os dois ciclos for são incrementados.
- Por fim, a função retorna o número de peças combinadas na jogada.

- direita

Objetivo:

- A função direita tem como objetivo mover todas as peças do tabuleiro para a direita.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de retorno:

- pecascombinadas, inteiro.

Variaveis auxiliares:

- pecascombinadas, inteiro;
- c, inteiro;
- l, inteiro;
- aux, inteiro.

Procedimento:

- A função inicia com dois ciclos for para percorrer a grelha.
- Dentro dos ciclos for existe uma condição if que verifica se o valor da posição da grelha é diferente ou igual a 0.
- Caso seja diferente, inicializa um ciclo while que verifica se o valor da posição na coluna à direita à correspondente é igual a zero ou igual à posição atual, iniciando uma condição if:
 - Caso o valor da posição na coluna à direita à correspondente seja igual à posição atual, o valor da posição atual é somado ao valor da posição da coluna à direita dessa mesma posição e a posição atual toma o valor de 0.
 - Caso o valor da posição da coluna à direita à correspondente seja igual a 0, o valor da posição da coluna atual será transportado para a posição da coluna à direita à correspondente e a posição atual toma o valor de 0.
- Caso o valor da posição da grelha seja igual a 0 os dois ciclos for são incrementados.
- Por fim, a função retorna o número de peças combinadas na jogada.

- esquerda

Objetivo:

- A função esquerda tem como objetivo mover todas as peças do tabuleiro para a esquerda.

Argumentos:

- sz, inteiro;
- grelha[sz][sz], inteiro.

Valor de retorno:

- pecascombinadas, inteiro.

Variaveis auxiliares:

- pecascombinadas, inteiro;
- c, inteiro;
- l, inteiro;
- aux, inteiro.

Procedimento:

- A função inicia com dois ciclos for para percorrer a grelha.
- Dentro dos ciclos for existe uma condição if que verifica se o valor da posição da grelha é diferente ou igual a 0.
- Caso seja diferente, inicializa um ciclo while que verifica se o valor da posição na coluna à esquerda à correspondente é igual a zero ou igual à posição atual, iniciando uma condição if:
 - Caso o valor da posição na coluna à esquerda à correspondente seja igual à posição atual, o valor da posição atual é somado ao valor da posição da coluna à esquerda dessa mesma posição e a posição atual toma o valor de 0.
 - Caso o valor da posição da coluna à esquerda à correspondente seja igual a 0, o valor da posição da coluna atual será transportado para a posição da coluna à esquerda à correspondente e a posição atual toma o valor de 0.
- Caso o valor da posição da grelha seja igual a 0 os dois ciclos for são incrementados.
- Por fim, a função retorna o número de peças combinadas na jogada.

Aspetos finais

Nos testes efetuados verificou-se que no modo automático ao ser executada a função `baixo()` encontra-se a somar peças que já foram somadas numa única jogada.

Para solucionar o problema foi tentado criar uma variável auxiliar que limitasse o movimento das peças para que não somasse as que já foram somadas nessa jogada, contudo não foi possível resolver o problema.