



DEEC

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES

TÉCNICO LISBOA

PROGRAMAÇÃO

(LEEC 21/22)

Enunciado do Projecto

Palavras Cruzadas

1. Introdução

O trabalho que se descreve corresponde ao projecto da UC de Programação para o MEEC em 2021/2022. O projecto a realizar corresponde à implementação de uma versão muito particular de um jogo de palavras cruzadas. O objectivo do projecto é promover e avaliar a prática dos conceitos e técnicas de programação estudadas no âmbito desta UC. O projecto deverá ser realizado ao longo do período acompanhando a matéria teórica lecionada nesta UC. Conforme descrito na secção de avaliação, haverá demonstrações parciais a realizar numa aula de laboratório e uma entrega final, que corresponde à entrega do projecto completo.

2. Descrição do Jogo

O jogo a implementar deverá ter vários modos de jogo, tanto para jogadores humanos, como implementação de estratégias de jogo pelo computador. De seguida descrevem-se o tabuleiro e as regras de jogo.

2.1. Tabuleiro de Jogo

O tabuleiro de jogo deve ser bidimensional e com dimensão dos seus eixos horizontal (linhas) e vertical (colunas) parametrizável. As coordenadas, linhas e colunas devem ser identificadas, respectivamente, por algarismos de 1 a 15 e de A a Z, conforme ilustrado na Fig. 1 (a), para um tabuleiro de dimensão 9x9 [linhas x colunas]. A dimensão mínima para as linhas e para as colunas é de 7, a dimensão máxima para as linhas é de 15 e para as colunas de 15. O número de linhas e colunas deve ser ímpar.

1	\$	3	\$
2	.	2	.	#	.	#	.	2	.	.	.
3	.	.	2	2	.	.	.
4	.	#	.	2	.	2	.	#	.	.	.
5	3	.	.	.	2	.	.	.	3	.	.
6	.	#	.	2	.	2	.	#	.	.	.
7	.	.	2	.	.	.	2
8	.	2	.	#	.	#	.	2	.	.	.
9	\$	3	\$
	A	B	C	D	E	F	G	H	I		

(a)

1	\$	3	\$
2	.	2	.	.	#	.	#	.	2	.	.	.
3	.	.	2	2	.	.	.
4	.	.	.	2	.	.	.	2
5	.	#	.	.	2	.	2	.	.	#	.	.
6	3	2	3	.
7	.	#	.	.	2	.	2	.	.	#	.	.
8	.	.	.	2	.	.	.	2
9	.	.	2	2	.	.	.
10	.	2	.	.	#	.	#	.	.	2	.	.
11	\$	3	\$
	A	B	C	D	E	F	G	H	I	J	K	

(b)

Fig. 1: Tabuleiro de jogo (a) 9x9 vazio, (b) 11x11 vazio.

Algumas casas do tabuleiro têm significados especiais:

. casa vazia, onde pode ser posta qualquer letra

casa proibida, onde não se pode pôr nenhuma letra

2 casa onde a letra lá colocada vale o dobro dos pontos

3 casa onde a letra lá colocada vale o triplo dos pontos

\$ casa onde a palavra lá colocada vale o dobro dos pontos

O tabuleiro pode ser lido de um ficheiro, caso em que a distribuição dos bónus é como estiver no ficheiro, isto é, as casas com significados especiais podem estar noutras posições e o tabuleiro pode já ter palavras pré-preenchidas.

Se o tabuleiro não for lido de um ficheiro, é inicializado como apresentado na Fig.1, de acordo com as seguintes regras (ver Fig. 1 (b) para o caso do tabuleiro 11x11):

- as casas nas diagonais do tabuleiro, excepto os cantos, são letra conta o dobro: '2'
- as casas a meio de cada aresta do tabuleiro têm letra conta triplo: '3'
- as casas 1 posição na diagonal em relação às casas letra conta triplo são proibidas: '#'
- os cantos do tabuleiro têm palavra conta dobro: '\$'

Pode-se formar palavras no tabuleiro, tanto na horizontal da esquerda para a direita, como na vertical de cima para baixo, mas nunca se consideram palavras na diagonal.

2.2. Língua de jogo

O jogo pode usar palavras em Inglês ou Português ou outra língua. Pode ser utilizado um dicionário para validar as palavras utilizadas. Mas não são permitidas letras acentuadas, nem palavras que não constem do dicionário, quando é utilizado um dicionário. Quando é utilizado um dicionário, este deve ser lido para a memória para uma consulta mais rápida. Dois exemplos de ficheiros de dicionário que podem ser utilizados são:

/usr/share/dict/words dicionário de Inglês

/usr/share/hunspell/pt_PT.dic dicionário de Português

Poderão também ser utilizados outros ficheiros de dicionário mais pequenos, pois o tempo de execução com ficheiros de dicionário grandes pode ser significativo.

São adoptadas as seguintes regras ao ler o dicionário de um ficheiro:

- cada linha do ficheiro só tem uma única palavra;
- a palavra termina quando aparecer espaço, tab, mudança de linha ou o carácter '/'
- se a palavra tiver alguma letra que não seja minúscula não acentuada, isto é, entre 'a' e 'z', é descartada;
- se a palavra tiver menos de 2 caracteres ou mais do que a maior dimensão do tabuleiro, é descartada.

Estas regras permitem simplificar o funcionamento do jogo, excluindo siglas, nomes próprios e possessivos em inglês.

Joga-se sempre com palavras escritas em minúsculas.

2.3. Posicionamento de palavras no Jogo e pontuação

Na primeira jogada, deve ser colocada uma palavra na horizontal, ocupando a casa central do tabuleiro.

Nas jogadas subsequentes, pode ser colocada uma palavra na horizontal ou vertical, mas é obrigatório usar pelo menos uma letra já existente no tabuleiro na nova palavra ou numa das novas palavras (no caso de serem formadas várias palavras). Ao formar palavras na horizontal, a palavra deve ler-se da esquerda para a direita. Ao formar palavras na vertical, a palavra deve ler-se de cima para baixo.

É definida uma notação para as jogadas, que corresponde à concatenação da letra da coluna, número da linha, 'H' para jogar na horizontal ou 'V' para jogar na vertical, um espaço e depois a palavra. Por exemplo para jogar a palavra "program" na horizontal na casa C5 escrever-se ia:

C5H program

Para simplicidade de digitação, podem-se aceitar minúsculas na leitura de jogadas para a coordenada horizontal e orientação da palavra, mas na escrita no écran ou em ficheiro devem-se escrever em maiúsculas. Ou seja, “c5h program” seria aceite para leitura e equivalente a “C5H program”.

Para indicar os pontos obtidos numa jogada, escreve-se a jogada seguida do número de pontos obtidos e da palavra “pontos”.

O tabuleiro resultante está na Fig. 2, ilustrando-se também os pontos obtidos.

1	\$.	.	.	3	.	.	.	\$
2	.	2	.	#	.	#	.	2	.
3	.	.	2	.	.	.	2	.	.
4	.	#	.	2	.	2	.	#	.
5	3	.	p	r	o	g	r	a	m
6	.	#	.	2	.	2	.	#	.
7	.	.	2	.	.	.	2	.	.
8	.	2	.	#	.	#	.	2	.
9	\$.	.	.	3	.	.	.	\$
	A	B	C	D	E	F	G	H	I

C5H program 19 pontos

Fig. 2: Tabuleiro com a primeira jogada.

A pontuação obtida numa jogada corresponde à soma dos pontos das novas palavras formadas. Os pontos obtidos por uma palavra correspondem à soma dos pontos de cada letra que forma a palavra, conforme a Fig. 3, multiplicada por eventuais bónus que as letras ou palavras tenham. Os bónus só são usados quando é colocada uma letra sobre uma casa do tabuleiro com bónus. Se depois for feita outra palavra aproveitando uma letra que já teve bónus, essa letra conta pelo seu valor, sem qualquer bónus adicional, visto que o bónus já foi usado e desapareceu do tabuleiro. Se, ao colocar uma palavra, houver múltiplos bónus, todos são usados cumulativamente. Por exemplo, se uma palavra apanhar um bónus de palavra conta dobro e outro de letra conta triplo, somam-se os pontos das letras triplicando a que tem bónus e o resultado da soma das letras depois de dobrado dá a pontuação final da palavra. Se uma palavra usar dois bónus de palavra conta dobro, a palavra contará o quádruplo.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	3	3	2	1	4	2	4	1	8	5	1	3	1	1	3	10	1	1	1	1	4	4	8	4	10

Fig. 3: Pontos a atribuir a cada letra, antes de considerar eventuais bónus.

Os pontos obtidos na primeira jogada, anteriormente referida, são:

$$p=3 + r=1 + o=1*2 + g=2 + r=1 + a=1 + m=3*3 = 19$$

É possível adicionar letras a uma palavra existente para formar novas palavras. Um exemplo seria a segunda jogada ser:

A5H reprogram

Neste caso, os bónus das letras que já estavam no tabuleiro já foram gastos, pelo que se obteriam 16 pontos apenas com esta jogada, visto que apenas o primeiro ‘r’ contaria o triplo.

Jogadas que saiam fora do tabuleiro, não usem uma letra já colocada, não coloquem letras novas, ou que ocupem casas proibidas não são válidas, devendo ser apresentada uma mensagem de erro ao utilizador para que ele tente jogar novamente. Exemplos de jogadas inválidas para a terceira jogada são, respectivamente:

I5V magical

A1H zebra

F5H gram

H4V game

Sempre que as letras de uma nova palavra formarem novas palavras na perpendicular, essas palavras têm de ser válidas e contam os respectivos pontos. No entanto, consoante os modos de jogo, as palavras na perpendicular podem não ser verificadas nem contabilizadas. Considere-se que as próximas 3 jogadas eram:

G5V room

G8H may

E9H paste

Esta última jogada forma, para além da palavra horizontal, três palavras verticais: “rooms”, “at” e “ye”, conforme ilustrado na Fig. 4. Os totais de pontos destas 3 jogadas são:

room: $1+1+1*2+3=7$ pontos

may: $3+1*2+4=9$ pontos

paste: rooms (7 pontos)+at (2 pontos)+ye (5 pontos*2) + paste (13*2) = 45 pontos

1	\$.	.	.	3	.	.	.	\$
2	.	2	.	#	.	#	.	2	.
3	.	.	2	.	.	.	2	.	.
4	.	#	.	2	.	2	.	#	.
5	r	e	p	r	o	g	r	a	m
6	.	#	.	2	.	2	o	#	.
7	.	.	2	.	.	.	o	.	.
8	.	2	.	#	.	#	m	a	y
9	\$.	.	.	p	a	s	t	e
	A	B	C	D	E	F	G	H	I

E9H paste 45 pontos

Fig. 4: Tabuleiro com várias jogadas.

2.4. Modos de Jogo

Os modos de jogo possíveis são 4:

1. Um jogador humano vai fazendo jogadas, sem validação por dicionário, sem restrição de letras que podem ser usadas para formar palavras, sem contabilizar e sem validar palavras na perpendicular.
2. O computador vai fazendo jogadas, sendo feita a validação das palavras jogadas através de um dicionário, sem restrição das letras que podem ser usadas para formar palavras, sem contabilizar e sem validar palavras na perpendicular.
3. O computador vai fazendo jogadas, sendo feita a validação das palavras jogadas através de um dicionário, e havendo restrição das letras que podem ser usadas para formar palavras, sem contabilizar e sem validar palavras na perpendicular.
4. O computador vai fazendo jogadas, sendo feita a validação das palavras jogadas através de um dicionário, e havendo restrição das letras que podem ser usadas para formar palavras, contabilizando-se palavras na perpendicular.

Nos modos 3 e 4, em que há restrição das letras que podem ser usadas para formar palavras, é dado um ficheiro com a sequência das letras que pode ser utilizada. O jogador tira um certo número de letras para a sua mão que pode usar para jogar no tabuleiro formando palavras. Depois da jogada, busca novas letras de forma a repor o número de letras na sua mão. Se já não for possível buscar mais letras, joga com as letras que tem na mão até elas acabarem. É necessário apresentar as letras que o jogador tem na mão antes e depois da sua jogada, ver Fig. 5. Depois da jogada, as letras utilizadas devem ter sido apagadas.

Nos jogos com jogadores humanos, o jogo termina quando o jogador não conseguir formar novas palavras porque já não tem letras na sua mão, ou quando o jogador quiser terminar, jogando “fim”. Nessa altura é-lhe apresentado o total de pontos que fez, ver Fig. 5.

Nos jogos em que o computador vai fazendo as jogadas, o jogo termina quando não for possível formar uma nova palavra no tabuleiro, quer por não haver letras na mão para jogar, quer por não haver sítio válido onde jogar. No fim do jogo, é apresentado o total de pontos feitos.

Por exemplo, se o ficheiro com a sequência das letras que pode ser utilizada tiver as letras “rogrampaingg” e for possível ter 7 letras na mão, para a primeira jogada, o jogador teria as primeiras 7 letras na mão, “rogramp”, e poderia jogar “A5H program”, obtendo 19 pontos. De seguida, buscaria até 7 letras, mas como só há mais 5 letras, ficaria com “aingg” na sua mão e poderia jogar “G3V gaming” obtendo 13 pontos e terminando o jogo porque não havia mais letras para buscar. O fim do jogo, incluindo o tabuleiro final está ilustrado na Fig. 5. Devem ser apresentadas as letras que o jogador tem na mão antes e depois de cada jogada.

```

Letras na mão: rogramp
A5H program 19 pontos
Letras na mão:
1 $ . . . 3 . . . $
2 . 2 . # . # . 2 .
3 . . 2 . . . g . .
4 . # . 2 . 2 a # .
5 p r o g r a m . 3
6 . # . 2 . 2 i # .
7 . . 2 . . . n .
8 . 2 . # . # g 2 .
9 $ . . . 3 . . . $
  A B C D E F G H I
Letras na mão: aingg
G3V gaming 13 pontos
Letras na mão:
Fim do jogo. Total de pontos obtidos: 32

```

Fig. 5: Tabuleiro após 2 jogadas com restrição das letras que podem ser usadas.

Na Fig. 6 apresenta-se um outro exemplo em que há 20 letras na mão para ilustrar como deve ser apresentada a utilização de letras em duas jogadas.

```

1 $ . . . 3 . . . $
2 . 2 . # . # . 2 .
3 . . 2 . . . 2 . .
4 . # . 2 . 2 . # .
5 c r a c k i n g 3
6 . # . 2 . 2 . # .
7 . . 2 . . . 2 . .
8 . 2 . # . # . 2 .
9 $ . . . 3 . . . $
  A B C D E F G H I
Letras na mão: agkuuutcidtbduvnwrnc
A5H cracking 28 pontos
Letras na mão:      uuut  dtbdv w n
1 $ . . . 3 . . . $
2 . 2 . # . # . 2 .
3 . . 2 . . . 2 . .
4 . # . 2 . j . # .
5 c r a c k i n g 3
6 . # . 2 . n . # .
7 . . 2 . . x 2 . .
8 . 2 . # . # . 2 .
9 $ . . . 3 . . . $
  A B C D E F G H I
Letras na mão: uuutdtbdvwnqjannxf
F4V jinx 27 pontos
Letras na mão: uuutdtbdv w q ann nf

```

Fig. 6: Exemplo de jogadas em que há restrição de letras a utilizar e há 20 letras na mão.

Caso o modo de jogo indique que não sejam contabilizadas palavras na perpendicular da palavra jogada, não se verifica no dicionário a validade das palavras na perpendicular da palavra jogada nem se contabilizam os pontos correspondentes. Assim, a jogada “E9H paste” na Fig.4 não seria permitida, visto que não usa uma letra já existente no tabuleiro e as três palavras verticais: “rooms”, “at” e “ye” não seriam validadas nem contabilizadas. Neste modo de jogo, um exemplo de jogada possível seria “C6H jumbo” que obteria 20 pontos, visto que as palavras perpendiculares (verticais) “pj”, “ru”, “om” e “gb” não são consideradas.

2.5. Algoritmo de Jogo do Computador

O algoritmo de jogo a implementar pelo computador é um algoritmo de força bruta que testa todo o dicionário em várias posições do tabuleiro, escolhendo a primeira jogada encontrada que resulta no máximo de pontos obtidos. Naturalmente, que o algoritmo tem limitações, sendo pouco eficiente para dicionários grandes ou tabuleiros complexos, podendo também não encontrar a sequência de jogadas que daria uma pontuação melhor no total do jogo. No entanto, convém que o algoritmo seja implementado exactamente como descrito, pois modificações no algoritmo resultam em jogadas diferentes e o programa será comparado pelos resultados produzidos.

Na primeira jogada, apenas a casa central do tabuleiro é analisada para pôr uma palavra na horizontal.

Nas jogadas seguintes, o algoritmo de jogo percorre todas as casas do tabuleiro de cima para baixo, da esquerda para a direita, percorrendo cada linha na horizontal antes de passar à linha seguinte. Sempre que uma casa tiver uma letra, ou se se considerar palavras na perpendicular e a casa não for proibida e imediatamente à esquerda estiver uma letra ou imediatamente à direita estiver uma letra, essa casa é passada ao algoritmo de análise de uma casa para pôr palavras na vertical. Tomando o exemplo do tabuleiro da Fig. 2, as casas que seriam analisadas para tentar pôr uma palavra na vertical

seriam sucessivamente B5, C5, D5, E5, F5, G5, H5, I5, conforme Fig. 7, por esta ordem, caso se considerem palavras na perpendicular e C5, D5, E5, F5, G5, H5, I5, por esta ordem, caso não se considerem palavras na perpendicular.

1	\$.	.	.	3	.	.	.	\$
2	.	2	.	#	.	#	.	2	.
3	.	.	2	.	.	.	2	.	.
4	.	#	.	2	.	2	.	#	.
5	3	.	p	r	o	g	r	a	m
6	.	#	.	2	.	2	.	#	.
7	.	.	2	.	.	.	2	.	.
8	.	2	.	#	.	#	.	2	.
9	\$.	.	.	3	.	.	.	\$
	A	B	C	D	E	F	G	H	I

Fig. 7: Tabuleiro com a primeira jogada e casas a analisar para tentar pôr uma palavra na vertical.

O algoritmo de jogo, em seguida, percorre novamente todas as casas do tabuleiro pela mesma ordem para tentar pôr palavras na horizontal. Sempre que uma casa tiver uma letra, ou se se considerar palavras na perpendicular e a casa não for proibida e imediatamente acima estiver uma letra ou imediatamente abaixo estiver uma letra, essa casa é passada ao algoritmo de análise de uma casa para pôr palavras na horizontal. Tomando o exemplo do tabuleiro da Fig. 2, as casas que seriam analisadas para tentar pôr uma palavra na horizontal seriam sucessivamente C4, D4, E4, F4, G4, I4, C5, D5, E5, F5, G5, H5, I5, C6, D6, E6, G5, I6, conforme Fig. 8, por esta ordem caso se considerem palavras na perpendicular e C5, D5, E5, F5, G5, H5, I5, por esta ordem, caso não se considerem palavras na perpendicular.

1	\$.	.	.	3	.	.	.	\$
2	.	2	.	#	.	#	.	2	.
3	.	.	2	.	.	.	2	.	.
4	.	#	.	2	.	2	.	#	.
5	3	.	p	r	o	g	r	a	m
6	.	#	.	2	.	2	.	#	.
7	.	.	2	.	.	.	2	.	.
8	.	2	.	#	.	#	.	2	.
9	\$.	.	.	3	.	.	.	\$
	A	B	C	D	E	F	G	H	I

Fig. 8: Tabuleiro com a primeira jogada e casas a analisar para tentar pôr uma palavra na horizontal.

O algoritmo de análise de uma casa é descrito para posicionar uma palavra na horizontal ocupando essa dada casa, devendo ser adaptado para posicionar palavras na vertical.

Começa-se com um tamanho de palavra o maior possível e depois vai-se diminuindo sucessivamente o tamanho da palavra até ao tamanho 2. Num tabuleiro de 9 colunas, testar-se-ia sucessivamente palavras de tamanho 9, 8, 7, 6, 5, 4, 3 e 2, por esta ordem.

Para cada tamanho de palavra, são testados sucessivamente vários deslocamentos da esquerda para a direita, desde que cubram a casa em análise e a palavra não saia do tabuleiro. Considerando como exemplo a casa central do tabuleiro (na coluna E), seriam testadas sucessivamente palavras com os tamanhos e posições indicados nos 24 passos da Fig. 9:

	coluna:	A	B	C	D	E	F	G	H	I
passo 1:	tamanho 9	1	2	3	4	5	6	7	8	9
passo 2:	tamanho 8	1	2	3	4	5	6	7	8	
passo 3:	tamanho 8		1	2	3	4	5	6	7	8
passo 4:	tamanho 7	1	2	3	4	5	6	7		
passo 5:	tamanho 7		1	2	3	4	5	6	7	
passo 6:	tamanho 7			1	2	3	4	5	6	7
passo 7:	tamanho 6	1	2	3	4	5	6			
passo 8:	tamanho 6		1	2	3	4	5	6		
passo 9:	tamanho 6			1	2	3	4	5	6	
passo 10:	tamanho 6				1	2	3	4	5	6
passo 11:	tamanho 5	1	2	3	4	5				
passo 12:	tamanho 5		1	2	3	4	5			
passo 13:	tamanho 5			1	2	3	4	5		
passo 14:	tamanho 5				1	2	3	4	5	
passo 15:	tamanho 5					1	2	3	4	5
passo 16:	tamanho 4	1	2	3	4					
passo 17:	tamanho 4		1	2	3	4				
passo 18:	tamanho 4			1	2	3	4			
passo 19:	tamanho 4				1	2	3	4		
passo 20:	tamanho 3	1	2	3						
passo 21:	tamanho 3		1	2	3					
passo 22:	tamanho 3			1	2	3				
passo 23:	tamanho 2	1	2							
passo 24:	tamanho 2		1	2						

Fig. 9: Ordem de teste de palavras na casa central de uma dada linha de um tabuleiro de 9 colunas

Para cada um dos passos seriam percorridas todas as palavras do dicionário com esse tamanho para verificar se uma palavra serve nessa posição e quais os pontos que se obteriam com essa jogada. Sempre que encontrar uma jogada possível, guarda-a, só a substituindo se encontrar outra jogada possível que dê mais pontos.

Para verificar se uma palavra pode ser jogada numa dada posição, é necessário que as várias casas onde a palavra seria colocada ou estejam vazias ou tenham lá a letra que essa palavra teria nessa posição, caso contrário a jogada é impossível. Se houver restrições das letras que podem ser usadas para formar palavras, as letras a colocar nas casas vazias têm de estar disponíveis na mão do jogador, caso contrário a jogada é impossível. Sempre que uma letra nova seria colocada e o modo de jogo indica que é para considerar palavras perpendiculares, é necessário verificar se se forma uma palavra válida na vertical a partir dessa letra. Se a palavra vertical for inválida, a jogada é impossível. Por exemplo, ao colocar a palavra “paste” em E9H na Fig. 4, formam-se 3 palavras nas verticais das letras ‘s’, ‘t’ e ‘e’, que são as palavras “rooms”, “at” e “ye”, que são palavras válidas. À medida que se formam palavras, têm de se contabilizar os pontos de acordo com os valores de cada letra conforme a Fig. 3 e os bônus que haja no tabuleiro.

O algoritmo para tentar pôr palavras na vertical é semelhante a este descrito, mas com a correspondente rotação no tabuleiro.

3. Interface de entrada/saída e parametrização do programa

3.1. Parametrização do programa

O programa deverá ser invocado na linha de comando da seguinte forma:

\$./palavras [OPTIONS]

'\$' é a prompt do Linux e "./" representa a directoria corrente

palavras designa o nome do ficheiro executável contendo o programa desenvolvido.

[OPTIONS] designa a possibilidade de o programa ser invocado com diferentes opções de funcionamento

As **opções** de funcionamento são identificadas sempre como strings começadas com o caractere '-', e podem aparecer por qualquer ordem. De seguida, descrevem-se as várias opções disponíveis:

-h	ajuda para o utilizador
-t lxc	dimensões do tabuleiro (linha x coluna).
-d filename	nome do ficheiro de dicionário a utilizar
-l filename	nome do ficheiro com letras a usar nas jogadas
-m 5-20	número de letras que um jogador pode ter na sua mão para jogar
-n nn	número de jogadas máximo a realizar
-i filename	define ficheiro com o tabuleiro a usar em alternativa a jogar num tabuleiro vazio
-j 1-4	modo de jogo 1 a 4
-o filename	define ficheiro onde escrever o tabuleiro final.
-r filename	define ficheiro para registo de todas as jogadas possíveis.

- A opção -h, quando invocada, deverá imprimir para stdout uma mensagem de ajuda de execução da linha de comandos. Para ver um exemplo deste tipo de mensagens, executar no terminal: "gedit -h".
- A opção -t deve indicar a dimensão do tabuleiro no formato linhaxcoluna, por exemplo, 9x13 (sem espaços). O valor pré-definido será 9x9.
- O valor pré-definido para a opção -d deve ser /usr/share/dict/words
- A opção -l (L minúsculo) indica o ficheiro de letras a usar se o modo de jogo indicar que se pretende jogar com restrição das letras que podem ser usadas para formar palavra. O nome do ficheiro por omissão é "letras.txt". O ficheiro terá uma sequência de letras minúsculas, podendo ter também mudanças de linha, que devem naturalmente ser ignoradas.
- A opção -m só tem significado se o modo de jogo indicar que se pretende jogar com restrição das letras que podem ser usadas para formar palavra.
- A opção -n indica o número máximo de jogadas a realizar, terminando depois o jogo. Por omissão, não há limite de jogadas.
- A opção -i indica o nome do ficheiro que deve ter o tabuleiro com que o jogo deve começar. O objetivo é poder continuar jogos anteriores. Por omissão, é usado um tabuleiro vazio como explicado na secção 2.1. A dimensão do tabuleiro é definida na opção -t.
- A opção -j indica o modo de jogo. O valor pré-definido será 1.
- A opção -o, indica o nome do ficheiro onde o programa deve escrever o tabuleiro final antes de terminar. O objetivo é poder continuar o jogo noutra altura. Por omissão, só se escreve em stdout.
- A opção -r indica o nome do ficheiro onde se registam todas as jogadas possíveis de acordo com o algoritmo de jogo. Por omissão, este registo não é feito. O registo só é feito nos modos em que o computador executa o algoritmo: -j 1, -j 2 e -j 3.

Exemplos válidos de invocação do programa:

Exemplo 1: `./palavras`

Exemplo 2: `./palavras -j 3 -d /usr/share/hunspell/pt_PT.dic`

Exemplo 3: `./palavras -j 4 -n 1 -i jogada3.txt -o jogada4.txt`

Exemplos inválidos de invocação do programa:

Exemplo 1: `./palavras -j 0`

Exemplo 2: `./palavras -t 9`

Exemplo 3: `./palavras -j`

Nota: Pode recorrer à função de biblioteca `getopt()` para efetuar a validação dos parâmetros de entrada ou implementar uma função para realizar essa validação. Consulte a página do manual com o comando “man 3 getopt” num terminal Linux ou numa pesquisa no Google.

No caso de invocação inválida do programa, o programa deve escrever uma mensagem de erro e terminar.

O formato dos ficheiros com tabuleiros é o indicado nas figuras apresentadas. Um ficheiro com um tabuleiro deve ter tantas linhas quantas o tabuleiro tem, mais a linha final com as letras das colunas. O número de linha e colunas é especificado na opção `-t`. Cada linha do ficheiro começa com o número da linha ocupando 2 caracteres (porque pode haver linhas maiores que 9), um espaço, seguido do conteúdo de cada coluna dessa linha separado por um espaço.

O ficheiro com o registo de jogadas possíveis é um ficheiro de texto. Tem uma linha com “Jogada” e o número da jogada, seguida de uma jogada possível por linha com os pontos correspondentes na mesma linha dessa jogada. Um exemplo, truncado, é apresentado na Fig. 10.

```
Jogada 1
A5H program 19 pontos
C5H program 19 pontos
B5H program 14 pontos
B5H pram 11 pontos
B5H prom 11 pontos
Jogada 2
G3V gaming 13 pontos
G3V gamin 11 pontos
A3V gaping 10 pontos
F3V again 9 pontos
F4V gag 9 pontos
```

Fig. 10: Exemplo de ficheiro com registo de jogadas (truncado).

Para o registo de jogadas possíveis, apenas necessário com a opção `-r`, o algoritmo de jogo do computador, em vez de registar apenas a jogada que obtém o máximo de pontos, deve guardar numa **lista criada dinamicamente** todas as possíveis jogadas analisadas. Depois de seleccionar a jogada que tem mais pontos, apaga da lista de jogadas analisadas todas as possíveis jogadas que tiverem menos de metade dos pontos máximos, correspondentes à jogada seleccionada, ficando assim com o top 50% das possíveis jogadas com mais pontos. Por exemplo, se o máximo de pontos é 19, todas as possíveis jogadas com 9 pontos ou menos são apagadas; se o máximo de pontos é 80, todas as jogadas com 39 pontos ou menos são apagadas. Estas possíveis jogadas são ordenadas por ordem decrescente de pontos, em caso de igualdade por posição no tabuleiro de cima para baixo, da esquerda para a direita, em caso de igualdade primeiro as jogadas horizontais e depois as verticais e em caso de igualdade por ordem alfabética da palavra. Depois de ordenada, as possíveis jogadas são escritas no

ficheiro. De seguida, a memória alocada dinamicamente para a lista deve ser libertada para que o algoritmo de jogo do computador possa passar à análise da próxima jogada.

3.2. Sugestões

Organize as palavras no dicionário por tamanho de palavra para não ter de percorrer sempre o dicionário todo.

4. Processo de Submissão

Os trabalhos submetidos deverão obrigatoriamente ser compilados com as seguintes opções:

-Wall -O3 -g

e correr na máquina virtual fornecida.

Deve ser submetido no Fénix um ficheiro ZIP com:

(1) código comentado com as funcionalidades indicadas no enunciado (ficheiros .h e .c);

(2) Makefile para gerar o executável;

Para além disso, será necessário preencher uma ficha de auto-avaliação no GoogleForms.

Por fim, o código deve ser estruturado de forma lógica em vários ficheiros (*.c e *.h). As funções devem ter um cabeçalho curto, mas explicativo e o código deve estar corretamente indentado e com comentários que facilitem a sua legibilidade.

É importante reforçar que certos modos de operação do programa serão avaliados de forma automática, pelo que é imperativo que o programa respeite a impressão para stdout/ficheiro, a leitura de stdin/ficheiro. A falha na execução dos mesmos poderá levar a uma penalização na nota final.

5. Processo de Avaliação

A avaliação do trabalho terá a seguinte distribuição de cotações:

- Modos de Jogo (J1 (**Lab4**), J2 (40%), J3 (10%) e J4 (10%)): 60%
- Gestão de Listas de Jogadas e Alocação Dinâmica (10%)
- Opções de Linha de Comando (10%)
- Comentários (10%)
- Qualidade do código (10%)

Avaliação oral (**realizada após a entrega do projecto**):

- Grupos com nota superior a 17 realizarão obrigatoriamente oral para defender a nota (a nota atribuída antes da oral será o ponto de partida e limite máximo da nota após a discussão oral)
- Grupos com nota entre 9.5 e 17 só realização oral se o docente de laboratório ou o avaliador de projecto considerar necessário.
- Grupos com nota menor do que 9.5 podem fazer oral para obter nota mínima de projecto.

6. Código de Honestidade Académica

Espera-se que os alunos conheçam e respeitem o Código de Honestidade Académica que rege esta disciplina e que pode ser consultado na página da cadeira. O projeto é para ser planeado e executado

por grupos de dois alunos e é nessa base que será avaliado. Quaisquer associações de grupos ou outras, que eventualmente venham a ocorrer, serão obviamente interpretadas como violação do Código de Honestidade Académica e terão como consequência a anulação do projeto aos elementos envolvidos.

Lembramos igualmente que a verificação de potenciais violações a este código é feita de forma automática com recurso a sofisticados métodos de comparação de código, que envolvem não apenas a comparação direta do código, mas também da estrutura do mesmo.