



DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

PROGRAMAÇÃO (MEEC 20/21)

**Enunciado do Projecto
Fase I: Entrega Intermédia**



MEEC War Game

1. Introdução

O trabalho que se descreve corresponde ao projecto da UC de Programação para o MEEC em 2020/2021. O projecto a realizar corresponde à implementação de uma versão muito particular do conhecido jogo da Batalha Naval. 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 semestre acompanhando a matéria teórica lecionada nesta UC. Conforme descrito na secção de avaliação, esta decorrerá em **duas fases**, a primeira, que se designa de entrega intermédia, a segunda e final, que corresponde à entrega do projecto completo, comentado e acompanhado de um curto relatório.

2. Descrição do Jogo

Pretende-se realizar uma implementação particular do conhecido jogo da Batalha Naval. O jogo a implementar deverá ter vários modos de jogo, tanto de jogador individual como de torneio entre vários jogadores. De seguida descrevem-se as peças, tabuleiros e regras de jogo.

2.1. Peças de Jogo

As peças do jogo são dos tipos indicados na Fig. 1, em que cada tipo de peça tem um número de quadrículas ou dimensão específico. O número e o tipo de peças de jogo a considerar em cada jogo varia em função do nível de dificuldade, conforme descrito adiante.

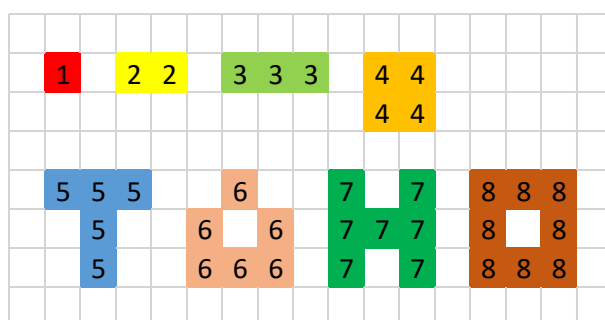


Fig. 1: Peças do jogo MEEC War Game.

2.2. 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. 2, para um tabuleiro de dimensão 9x9 [linhas x colunas]. A dimensão mínima para as linhas e para as colunas é de 9, a dimensão máxima para as linhas é de 15 e para as colunas de 24 (em ambos os eixos, o tabuleiro só pode assumir valores múltiplos de 3).

Na Fig. 2 ilustra-se o tabuleiro de jogo onde são apresentados os algarismos nas posições onde estão as peças.

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

9	4	4	-	-	-	-	-	4	4
8	4	4	-	3	3	3	-	4	4
7	-	-	-	-	-	-	-	-	-
6	-	-	-	5	-	-	-	2	-
5	-	1	-	5	5	5	-	2	-
4	-	-	-	5	-	-	-	-	-
3	6	6	-	-	-	-	-	8	8
2	6	-	6	-	2	-	-	8	-
1	6	6	-	-	2	-	-	8	8
	A	B	C	D	E	F	G	H	I

Fig. 2: Exemplos de tabuleiro de jogo (a) vazio; (b) completo.

2.3. Posicionamento das Peças de Jogo

O posicionamento das peças no tabuleiro de jogo deve seguir alguns critérios cuja aplicação dependerá do nível de jogo.

Restrição 0) - As peças só podem ser colocadas em quadrículas de 3x3, conforme ilustrado na Fig. 2. Uma peça não pode ocupar mais do que uma quadrícula de 3x3 definida no tabuleiro. As quadrículas de 3x3 do tabuleiro definem-se avançando no eixo vertical e no eixo horizontal de 3 em 3. Dado que o tabuleiro tem sempre dimensão múltipla de 3 em ambos os eixos, pode ser sempre construído com base num número inteiro de quadrículas de 3x3 em ambas as direções.

Restrição 1) - As peças nunca podem ter arestas ou vértices de contacto com outras peças. A Fig. 3 a) contém um exemplo de tabuleiro inválido, enquanto a Fig. 3b) contém um tabuleiro semelhante, mas válido.

9	-	4	4	-	-	-	-	-	-
8	-	4	4	3	3	3	-	-	-
7	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-
3	6	6	-	-	-	-	-	-	-
2	6	-	6	-	-	-	-	-	-
1	6	6	-	-	-	-	-	-	-
	A	B	C	D	E	F	G	H	I

Tabuleiro Inválido

9	-	4	4	-	3	-	-	-	-
8	-	4	4	-	3	-	-	-	-
7	-	-	-	-	3	-	-	-	-
6	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-
3	6	6	-	-	-	-	-	-	-
2	6	-	6	-	-	-	-	-	-
1	6	6	-	-	-	-	-	-	-
	A	B	C	D	E	F	G	H	I

Tabuleiro válido

Fig. 3: Exemplos de tabuleiro de jogo (a) inválido; (b) válido.

Restrição 2) - O número máximo de peças por tabuleiro está limitado a: $\text{linhas} * \text{colunas} / 9$. No caso de o número de peças ser inferior ao limite, devem-se considerar, nos espaços restantes, matrizes de 3x3 vazias.

2.4. Modos de Jogo

Os modos de jogo possíveis são 3:

1. O jogador tenta descobrir a posição das peças do computador; [Entrega Intermédia]
2. O computador tenta descobrir a posição das peças do jogador; [Entrega Intermédia]
3. O programa implementa um torneio entre diferentes perfis de jogo do computador. [Entrega Final]

3. Implementação

Para efeito de implementação, cada grupo deve tomar as suas opções em termos de representação de dados, estruturação do código em funções e em módulos.

3.1. Peças de Jogo

Cada grupo deve decidir a forma mais adequada para representar as peças respeitando a identificação global e o identificador de variante associado a cada peça conforme descrito no anexo (no fim deste documento). Na Fig. 4, ilustra-se o que se designa por diferentes instâncias ou variantes da mesma peça.

1	-	-
-	-	-
-	-	-

-	1	-
-	-	-
-	-	-

Fig. 4: Exemplo de 2 instâncias da peça 1.

Na Fig. 5, apresenta-se um exemplo de uma instância possível para cada um dos 8 tipos de peças.

-	-	-
-	-	-
-	1	-

-	-	-
2	2	-
-	-	-

-	3	-
-	3	-
-	3	-

-	-	-
4	4	-
4	4	-

5	-	-
5	5	5
5	-	-

-	6	-
6	-	6
6	6	6

7	7	7
-	7	-
7	7	7

8	8	8
8	-	8
8	8	8

Fig. 5: Exemplo de uma instância para cada um dos tipos de peça disponíveis.

Restrição 3) - Num jogo, segue-se a regra de **nunca ter mais peças de um tipo de maior dimensão do que peças de um tipo de menor dimensão**. Por exemplo:

- 3 peças do tipo 1 e 2 peças do tipo 2 – **válido**
- 2 peças do tipo 1 e 2 peças do tipo 2 – **válido**
- 2 peças do tipo 1 e 3 peças do tipo 2 – **inválido**

Restrição 4) - Mais, o **número total de peças não deve exceder 50% do número de matrizes 3x3 num tabuleiro**. Por exemplo:

- 4 peças num tabuleiro de 9x9 quadrículas – **válido**
- 5 peças num tabuleiro de 9x9 quadrículas – **inválido (devido à restrição 4)**
- 20 peças num tabuleiro de 15x24 quadrículas – **válido**
- 21 peças num tabuleiro de 15x24 quadrículas – **inválido (devido à restrição 4)**

3.2. Tabuleiro de Jogo

Cada grupo deve decidir a forma mais adequada para representar o tabuleiro de jogo, tendo de respeitar a identificação de linhas e colunas (ver Fig. 2). As linhas identificam-se, de cima para baixo, em numeração decrescente, começando com o número de linhas e acabando em 1. As colunas identificam-se por ordem alfabética (alfabeto internacional em maiúsculas), da esquerda para a direita, começando no A. A dimensão mínima do tabuleiro é de 9 linhas por 9 colunas e a dimensão máxima é de 15 linhas e 24 colunas. O número de linhas e colunas deverá ser sempre múltiplo de 3.

Inicialmente, na fase 1 [Entrega intermédia] a **alocação de memória** para representar o tabuleiro deve ser feita de forma **estática**.

3.3. Posicionamento das Peças de Jogo

O posicionamento das peças deve ter vários modos **modo_pn**.

- **modo_p1 [Entrega Intermédia]:** Coloca peças de tipo e variante aleatória no tabuleiro **sem cumprir as restrições 3 e 4 apresentadas na secção 3.1**, seguindo os passos:
 - **P0:** inicializa-se a **coordenada de posicionamento** no canto superior esquerdo do tabuleiro e **avança para P2**. Por exemplo, para um tabuleiro de 9x9 deve iniciar na coordenada A9.
 - **P1:** atualiza a coordenada de posicionamento e **avança para P2**. A atualização da posição deve ser feita prioritariamente na horizontal, isto é, mantendo a linha e deslocando 3 colunas para a direita; caso não seja possível, desce 3 linhas e coloca-se na 1ª coluna. Por exemplo, num tabuleiro de 9x9, se estiver na posição A9 avança para a D9, mas se estiver na posição G9 avança para a A6.
 - **P2:** escolhe-se aleatoriamente o **tipo e variante da peça** a posicionar **ou uma matriz de 3x3 vazia** e **avança para P3**.
 - **P3:** verifica se pode posicionar a matriz de 3x3 na posição indicada pela coordenada de posicionamento (**deve cumprir as restrições 0 e 1**), se sim coloca e **avança para P4**, se não **volta para P2** (se ao fim de **3 tentativas** não conseguir colocar a peça gerada, então deve colocar a peça com identificador global 5, ou seja, peça tipo 1 centrada).
 - **P4:** verifica se chegou ao fim do tabuleiro (deve cumprir a restrição 2), se sim conclui o posicionamento de peças registando o número de peças de cada tipo, se não regressa a P1.

Nota: Neste modo modo_p1, não se aplicam as restrições 3 e 4.

A Fig. 6 apresenta um exemplo de um posicionamento possível usando este modo de posicionamento (modo_p1).

9	4	4	-	-	-	-	-	4	4
8	4	4	-	3	3	3	-	4	4
7	-	-	-	-	-	-	-	-	-
6	-	-	-	5	-	-	-	2	-
5	-	1	-	5	5	5	-	2	-
4	-	-	-	5	-	-	-	-	-
3	6	6	-	-	-	-	8	8	8
2	6	-	6	-	2	-	8	-	8
1	6	6	-	-	2	-	8	8	8
	A	B	C	D	E	F	G	H	I

Fig. 6: Tabuleiro possível no **modo_p1** (mas não no modo_p2 uma vez que tem mais peças do tipo 2 do que do tipo 1, e mais do tipo 4 do que do tipo 3, o que viola a restrição 3)

- **modo_p2 [Entrega Intermédia]:** Fornecidos o número e tipo de peças a posicionar (**mas não a variante das peças**), e a dimensão do tabuleiro, devem ser seguidos os seguintes passos:

- **P0:** calcula o número de matrizes vazias e verifica se o número de peças ($n_{\text{peças}}$) a posicionar pode ser colocado no tabuleiro de acordo com seguinte regra (**para além de respeitar as restrições 3 e 4**):

$$n_{\text{peças}} \leq (\text{linhas} * \text{colunas} / 9) / 2$$

Desta regra fica implícito que o **número de matrizes vazias é maior ou igual ao número de matrizes ocupadas**.

- **P1:** inicializa-se a **coordenada de posicionamento** no canto superior esquerdo do tabuleiro e **avança para P3**. Por exemplo, para um tabuleiro de 9x9 deve iniciar na coordenada A9.
- **P2:** atualiza a coordenada de posicionamento e **avança para P3**. A atualização da posição deve ser feita prioritariamente na horizontal, isto é, mantendo a linha e deslocando 3 colunas para a direita, caso não seja possível desce 3 linhas e coloca-se na 1ª coluna. Por exemplo, num tabuleiro de 9x9, se estiver na posição A9 avança para a D9, mas se estiver na posição G9 avança para a A6.
- **P3:** escolhe-se aleatoriamente o tipo de peça a posicionar, de entre as peças ainda não posicionadas, ou uma matriz de 3x3 vazia. De seguida, no caso de uma matriz não vazia, seleciona aleatoriamente uma variante da peça a posicionar. **Avança para P4**.
- **P4:** verifica se pode posicionar a matriz de 3x3 na posição indicada pela coordenada de posicionamento (**deve cumprir as restrições 0 e 1**), se sim coloca e avança para P5, se não testa todas as variantes da peça seguindo a ordem do anexo, caso não seja possível posicionar nenhuma variante regressa a P3, exceto no caso de ter realizado 8 tentativas de colocar uma nova peça (regressos a P3, não deve tentar colocar uma peça de um tipo já explorado em tentativas anteriores) em que regressa a P1 (ao fim de 1000 regressos a P1, devolve -1 indicando que não conseguiu gerar o tabuleiro)
- **P5:** verifica se chegou ao fim do tabuleiro, se sim conclui o posicionamento de peças, se não regressa a P2.

NOTA: No posicionamento modo_p2, todas as restrições são tidas em consideração.

```

15 - - - 7 7 7 - - - - - - - - - 6 6 - 1 -
14 - - - 7 - - - - - - 3 3 3 - - - 6 - 6 - -
13 - - - 7 7 7 - - - - - - - - - 6 6 - - -
12 - - - - - - 6 - - - - - - - - - 1 - - - - 2 2
11 - - - - - 6 - 6 - - - - - - - - - 2 - - -
10 - - - - - 6 6 6 - - - - - - - - - 2 - - -
9 8 8 8 - - - - - - - - - 8 8 8 - - - - - -
8 8 - 8 - - - - - - 4 4 - 8 - 8 - - - - - -
7 8 8 8 - - - - - - 4 4 - 8 8 8 - 2 2 - - - -
6 - - - - - - - - - - - - - - - - - - 3 3 3
5 4 4 - - - - - - - - - - - - - - - - - -
4 4 4 - - - - - - - - - - - - - - - - - -
3 - - - 5 - - - - - 5 - - - 3 - 4 4 - - - 7 - 7
2 - - - 5 - - - - - 5 - - - 3 - 4 4 - - - 7 7 7
1 - - - 5 5 5 - 1 - 5 5 5 - 3 - - - - - 7 - 7
  A B C D E F G H I J K L M N O P Q R S T U V W X

```

Fig. 7: Exemplo de tabuleiro de jogo de 15x24, com a combinação mais complexa permitida no modo_p2.

3.4. Modos de Disparo e Estratégia do Computador

O modo de disparo impõe restrições relativamente à sequência das coordenadas de disparo. Cada jogada corresponde a um disparo numa dada célula. Os vários modos de disparo a implementar para o modo em que o computador joga são os seguintes:

- **modo_d1 [Entrega Intermédia]:**
 - **P1:** Escolhe aleatoriamente as coordenadas para o disparo, que não devem coincidir com disparos já realizados. A Fig. 8 demonstra um exemplo deste modo de disparo, em que o computador fez a sequência de disparos: B8, B2, C6, D6, H5 e H7. Avança para P2.
 - **P2:** Verifica se atingiu completamente todas as peças ou se explorou completamente o tabuleiro, se sim, conclui o jogo, caso contrário regressa a P1.

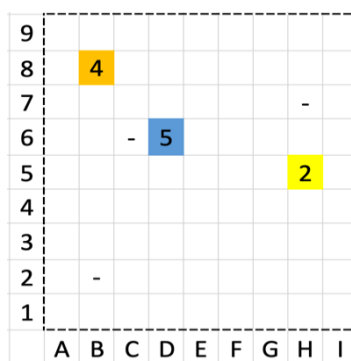


Fig. 8: Exemplo de disparos no **modo_d1**

- **modo_d2 [Entrega Intermédia]:**
 - **P1:** Inicializa-se a coordenada de disparo, na célula central, da matriz de células 3x3 no canto superior esquerdo do tabuleiro. Por exemplo, para um tabuleiro de 9x9, a coordenada de disparo inicial seria a B8. Avança para P2.
 - **P2:** Em cada matriz 3x3, os disparos são efetuados de acordo com a sequência de disparos indicada na Fig. 9. Neste modo, caso um barco/navio seja eliminado antes de se concluir o padrão, excluem-se as restantes células que pertencem à matriz de 3x3 e avança para P3.
 - **P3:** Verifica se atingiu completamente todas as peças, se sim, conclui o jogo, caso contrário, avança para P4.
 - **P4:** A matriz seguinte, a explorar, deve ser a matriz à direita. Caso não existam mais matrizes 3x3 nessa linha, o cursor deve ser posicionado no centro da matriz mais à esquerda imediatamente abaixo das matrizes já exploradas. Por exemplo, num tabuleiro de 9x9, após explorar a matriz superior mais à direita, a coordenada de disparo deve ser B5. De seguida, passa para P2.

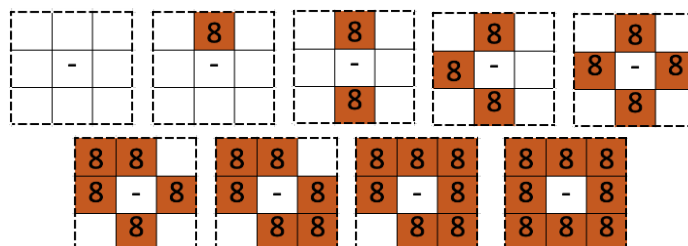


Fig. 9: Ordem dos disparos para cada matriz 3x3

Na Fig. 10, ilustra-se o modo 2 de disparo. Neste caso, pode-se verificar que a sequência usada foi:

- B8, B9, B7, A8, C8, A9 (peça eliminada, passamos à matriz 3x3 seguinte)
- E8, E9, E7, D8, F8, D9 (matriz 3x3 seguinte)
- Repete até ao fim do tabuleiro

9	4	4		2	2			
8	4	4	-	-	-	-	-	
7	-			-				
6								
5								
4								
3								
2								
1								
	A	B	C	D	E	F	G	H

Fig. 10: **modo_d2** aplicado a um tabuleiro

- **modo_d3 [Entrega Intermédia]:** Este modo de disparo é uma “evolução” do **modo_d2**. À semelhança do modo anterior, pretende-se que o algoritmo comece a efetuar os disparos na matriz de células 3x3 no canto superior esquerdo de acordo com o padrão identificado na Fig. 9. Após destruir o navio (peça) deve-se passar para a matriz seguinte, **mas eliminando as posições adjacentes ao navio destruído**, de acordo com o exemplo na Fig. 11 a). Por exemplo, a partir do instante em que o navio do tipo 4 é eliminado, as células indicadas com um X não podem ser usadas para efetuar disparos (**relembre a restrição 1**). Nota, deve concluir quando todas as peças tenham sido atingidas na totalidade.

Na Fig. 11 a), b) c) é apresentado um exemplo respetivo a este modo.

9	-	-	X	X				
8	-	4	4	X				
7	X	4	4	X				
6	X	X	X	X				
5								
4								
3								
2								
1								
	A	B	C	D	E	F	G	H

Região onde já não pode disparar

9	-	-	X	X	-	X		
8	-	4	4	X	2	X		
7	X	4	4	X	2	X		
6	X	X	X	X	X	X		
5								
4								
3								
2								
1								
	A	B	C	D	E	F	G	H

A região adjacente onde não pode disparar cresceu

9	-	-	X	X	-	X	-	1
8	-	4	4	X	2	X	-	-
7	X	4	4	X	2	X	-	-
6	X	X	X	X	X	X		
5								
4								
3								
2								
1								
	A	B	C	D	E	F	G	H

A região adjacente a 1 já foi testada (segundo o padrão mencionado)

Fig. 11: a) Exemplo de possibilidades de disparos que deixam de poder ser consideradas
b) Exemplo em que a região inválida cresce
c) Exemplo em que o tamanho da região inválida não cresce

Neste caso, pode-se verificar que a sequência de disparos foi:

- B8, B9, B7, A8, C8, A9, C7 (peça eliminada, passando à matriz seguinte, descartando células em redor do navio afundado)
- E8, E9, E7 (peça eliminada, passando à matriz seguinte, descartando mais células em redor do navio afundado)
- H8, H9, H7, G8, I8, G9, I9 (navio afundado, passando à matriz seguinte, mas a região inválida não cresce, pois não há células adjacentes ao submarino por explorar).

A partir de agora as células indicadas com uma cruz (região inválida) não podem sofrer disparos, logo, de acordo com a Fig. 12, os próximos disparos efetuados serão:

- B5, B4, A5, C5, C4, A4

Como se pode verificar através desta sequência, as células A6, B6 e C6 não foram alvos de disparo, apesar de pertencerem ao padrão de disparos a seguir. Tal deve-se ao facto destas mesmas células pertencerem à região inválida desde que o barco do tipo 4 foi eliminado.

9	-	-	X	X	-	X	-	-	1
8	-	4	4	X	2	X	-	-	-
7	X	4	4	X	2	X	-	-	-
6	X	X	X	X	X	X	-	-	-
5	-	-	-	-	-	-	-	-	-
4	1	-	-	-	-	-	-	-	-
3	X	X	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-
	A	B	C	D	E	F	G	H	I

Fig. 12: Exemplo de disparos efetuados na quarta matriz 3x3

Nota: os símbolos 'X' usados são meramente ilustrativos das regiões que se tornam inválidas para disparos. Não precisam de ser usados no programa, mas podem ser usados como auxiliar no desenvolvimento do algoritmo.

3.5. Modos de Jogo

Os modos de jogo dividem-se em:

- **modo_j0 [Entrega Intermédia]:** Neste modo, o programa gera 1 tabuleiro da dimensão pretendida e com o modo de posicionamento especificado.
- **modo_j1 [Entrega Intermédia]:** O jogador tenta acertar nas peças escondidas no tabuleiro do computador. Neste modo, o jogador deve especificar as coordenadas de disparo indicando colunalinha, por exemplo: C12, e o programa responde com o caractere '-' ou com o dígito correspondente à peça atingida de '1' a '8'. Assim que o jogador acertar na totalidade das coordenadas correspondentes às peças no tabuleiro, o programa deve dar o jogo por concluído, indicando o número de tentativas realizadas e o tempo de jogo em segundos.
- **modo_j2 [Entrega Intermédia]:** O computador tenta descobrir e afundar as peças no tabuleiro do jogo usando um dos modos de disparos implementado. Neste modo, o programa deve especificar as coordenadas de disparo indicando colunalinha, por exemplo: A6, e o jogador responde com o caractere '-' ou com o dígito correspondente à peça atingida de '1' a '8'. Assim que o programa acertar na totalidade das coordenadas correspondentes às peças no tabuleiro, deve dar o jogo por concluído, indicando o número de tentativas realizadas e o tempo de jogo em segundos.

3.6. Modos de Execução

A interface com o utilizador está decomposta entre entradas (inputs) e saídas (outputs) conforme se descreve de seguida.

3.6.1. Invocação do Programa

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

prog\$./wargame [OPTION]

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

[OPTION] 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 '-'. De seguida, descrevem-se as várias opções disponíveis:

-h	ajuda para o utilizador
-t	dimensões do tabuleiro (linha x coluna).
-j	modo de jogo (0 a 2)
-p	modo de posicionamento de peças pelo computador (1 a 2)
-d	modo de disparo de peças pelo computador (1 a 3)
-1	número de peças tipo 1 (mínimo 1)
-2	número de peças tipo 2
-3	número de peças tipo 3
-4	número de peças tipo 4
-5	número de peças tipo 5
-6	número de peças tipo 6
-7	número de peças tipo 7
-8	número de peças tipo 8

- 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: "vim -h".
- A opção -t, deve indicar a dimensão do tabuleiro no formato linhaxcoluna, por exemplo, 9x12 (sem espaços). O valor pré-definido será 9x9.
- A opção -j, indica o modo de jogo. O valor pré-definido será 0.
- A opção -p, indica o modo de posicionamento. O valor pré-definido será 1.
- A opção -d, indica o modo de disparo. O valor pré-definido será 1.
- As opções -[1-8] indicam o número de peças de cada tipo.
 - No modo de jogo 0 e 1, apenas são válidas quando se considera o posicionamento modo_p2. Neste caso, se a restrição das peças definida nas secções anteriores não é cumprida, o programa deverá terminar e imprimir para stdout -1. Caso o modo_p1 seja definido e ao mesmo tempo opções -[1-8], deverá ser impresso em stdout a mensagem de ajuda (previamente definida para -h).

- No modo de jogo 2, estas opções são utilizadas para indicar ao jogador o número de peças que deve incluir no tabuleiro e para o programa saber quando concluiu o jogo.
- As opções de -d [1-3] apenas são válidas quando o modo de jogo é modo_j2. Caso contrário deverá ser impresso em stdout a mensagem de ajuda (previamente definida para -h).

Exemplos válidos de invocação do programa:

Exemplo 1: `./wargame -j 1 -p 1`

Exemplo 2: `./wargame -j 2 -p 1 -d 3`

Exemplo 3: `./wargame -d 3 -j 1 -p 2 -1 5 -2 3`

Exemplos inválidos de invocação do programa:

Exemplo 1: `./wargame -j 1 -p 1 -d 3`

Exemplo 2: `./wargame -j 1 -d 2`

Exemplo 3: `./wargame -j 2 -d 1 -1 5 -2 4 -3 5`

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.

3.6.2. Execução do Programa

O programa pode-se executar nos diferentes modos com interface para o utilizador na janela de terminal ou em modo de teste automático, conforme descrito de seguida.

Modo de Jogo 0

Este modo será utilizado para validar o preenchimento dos tabuleiros de jogo gerados pelo programa. Neste modo o programa deve ser chamado do seguinte modo:

prog\$./wargame -t linhaxcoluna -j 0 -p [1 ou 2] [-1 a -8]

A opção -t deverá ser obrigatoriamente especificada para indicar a dimensão do tabuleiro a considerar. A opção -j deverá ser colocada a 0. A opção -p pode ser 1 ou 2, no caso de ser 2 devem especificar-se o número de peças de cada tipo (a não especificação deve assumir que esse tipo tem 0 peças). Neste modo, deve ser gerado 1 tabuleiro com as características indicadas e enviadas para o stdout. Os tabuleiros devem ser enviados para o stdout no seguinte formato por linha: a primeira linha deve ser indicada a dimensão do tabuleiro e o número de peças de cada tipo (de 1 a 8) separados por espaço no formato indicado, as 2 primeiras colunas para apresentar a ordenada, seguida de uma coluna vazia (espaço) e as seguintes para representar o conteúdo do tabuleiro (separando as colunas por 1 espaços), a última linha deve ser a linha das abcissas alinhadas com as respectivas colunas.

```

12x9 4 3 1 1 1 1 1 0
12 - - - - 2 - - 3 -
11 - 1 - - 2 - - 3 -
10 - - - - - - - 3 -
9 5 5 5 - - - - - -
...
1 4 4 - - 2 - - 5 -
  A B C D E F G H I

```

Fig. 13: Output do programa no modo de jogo 0.

Modo de Jogo 1

Este modo será utilizado para validar as respostas do programa aos disparos do jogador. Neste modo o programa deve ser chamado do seguinte modo:

prog\$./wargame -t linhaxcoluna -j 1 -p [1 ou 2] [-1 a -8]

A opção -t deverá ser obrigatoriamente especificada para indicar a dimensão do tabuleiro a considerar. A opção -j deverá ser colocada a 1. A opção -p pode ser 1 ou 2, no caso de ser 2 devem especificar-se o número de peças de cada tipo (a não especificação deve assumir que esse tipo tem 0 peças).

O jogador deve especificar as coordenadas de disparo no formato *colunalinha*, por exemplo, C12, e o programa responde com o caractere '-' ou com o dígito correspondente à peça atingida de '1' a '8'. Assim que o jogador acertar na totalidade das coordenadas correspondentes às peças no tabuleiro, o programa deve dar o jogo por concluído, indicando o número de tentativas realizadas e o tempo de jogo em segundos. Por fim, o programa deve apresentar o tabuleiro que tinha sido gerado. A interface deve ter o formato indicado na fig. 14. A interface pode ou não ter informação sobre o jogo, se tiver deve ser no começo devendo essas linhas serem iniciadas por '*'. Seguidamente, deve ser escrita uma linha com a dimensão do tabuleiro e o número de peças de cada tipo (de 1 a 8), conforme indicado na fig.14, e sempre separado por espaços. De seguida, e alternadamente, por cada nova linha deve ser lida a coordenada de disparo e o resultado do disparo. No fim de jogo deve ser escrita uma linha com: "Fim de Jogo: x jogadas em y segundos". Por fim, deve ser apresentado o tabuleiro no formato especificado para o modo de jogo 0.

```

* =====
* Modo de Jogo 1
* Insira as Coordenadas de Disparo
* =====
12x9 4 3 1 1 1 1 1 0
A11
-
C9
5
...
Fim de Jogo: 75 jogadas em 125 segundos
12 - - - 2 - - 3 -
11 - 1 - - 2 - - 3 -
10 - - - - - - 3 -
 9 5 5 5 - - - - -
...
 1 4 4 - - 2 - - 5 -
   A B C D E F G H I

```

Fig. 14: Output do programa no modo de jogo 1.

Modo de Jogo 2

Este modo será utilizado para validar os modos de disparo do programa. Neste modo o programa deve ser chamado do seguinte modo:

prog\$./wargame -t linhaxcoluna -j 2 -d [1 ou 3] [-1 a -8]

A opção -t deverá ser obrigatoriamente especificada para indicar a dimensão do tabuleiro a considerar. A opção -j deverá ser colocada a 1. A opção -d pode ser 1 ou 3. Em qualquer caso, deve ser especificado o número de peças de cada tipo (a não especificação deve assumir que esse tipo tem 0 peças). De seguida, e alternadamente, por cada nova linha deve ser escrita a coordenada de disparo por parte do programa e a resposta por parte do jogador. No fim de jogo, assim que o programa tiver acertado completamente em todas as peças ou tenha realizado o número máximo de disparos (igual à dimensão do tabuleiro) deve ser escrita uma linha com: “Fim de Jogo: x jogadas em y segundos”. Por fim, deve ser apresentado o tabuleiro no formato especificado para o modo de jogo 0.

```

* =====
* Modo de Jogo 2
* Crie um tabuleiro com as características indicadas
* Responda aos disparos do programa
* =====
12x9 4 3 1 1 1 1 1 0
A11
-
C9
5
...
Fim de Jogo: 70 jogadas em 77 segundos
12 - - - - 2 - - 3 -
11 - 1 - - 2 - - 3 -
10 - - - - - - - 3 -
 9 5 5 5 - - - - -
...
 1 4 4 - - 2 - - 5 -
   A B C D E F G H I

```

Fig. 15: Output do programa no modo de jogo 2.

Nota: Por fim, no caso de invocação do programa com erro (especificação errada de parâmetros) deve retornar -1 seguido de uma mensagem, na mesma linha, indicando o tipo de erro.

4. Processo de Submissão

Os trabalhos submetidos deverão obrigatoriamente ser compilados com as seguintes opções
-Wall -std=c11 -O3 e correr na máquina virtual fornecida.

- **Entrega Intermédia (23 de abril de 2021)**

Submissão no Fénix do: (1) código comentado com as funcionalidades indicadas para a entrega intermédia; (2) Descrição da estrutura de dados utilizada para representar os tabuleiros e as peças (máx: 1 página); (3) ficha de auto-avaliação a preencher no GoogleForms.

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, a leitura de stdin. A falha na execução dos mesmos poderá levar a uma penalização na nota final.

5. Avaliação

A avaliação do trabalho terá a seguinte distribuição de cotações (**Entrega Intermédia**):

- Modos de Jogo (J0 (30%), J1 (10%) e J2 (30%)): 70%
- Comentários: 10%
- Qualidade do código: 20%
- Relatório ou guia de implementação: Obrigatório, cotação embebida nos tópicos em avaliação.
- Avaliação oral a todos os grupos (**realizada após a entrega final**)
 - A avaliação obtida resultante da ponderação do projecto intermédio (1/3) e do projeto final (2/3), antes da discussão, será o ponto de partida e limite máximo da nota após a discussão oral.

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.

7. Anexo: Biblioteca de Posicionamento das Peças

Descrevem-se de seguida as matrizes de 3x3 correspondentes às 42 representações das 8 peças que devem ser consideradas. Conforme descrito no enunciado as peças devem poder ser referenciadas tanto pelo seu tipo (valor sobre o canto superior esquerdo, de 1 a 8) e respetiva variante (valor sobre o quadrado central da 1ª linha da matriz, de 1 a nv (nº máximo de variantes)) como pela sua sequência absoluta (valor sobre o canto superior direito, de 1 a 42).

Coordenadas a considerar na matriz de peças

