

Jogo - Batalha Naval

Este exercício consiste no desenvolvimento de um jogo de Batalha Naval reduzido, que contemplará as peças demonstradas na Figura 1, sendo constituídas por um tabuleiro de 15 x 15 posições (Figura 1).

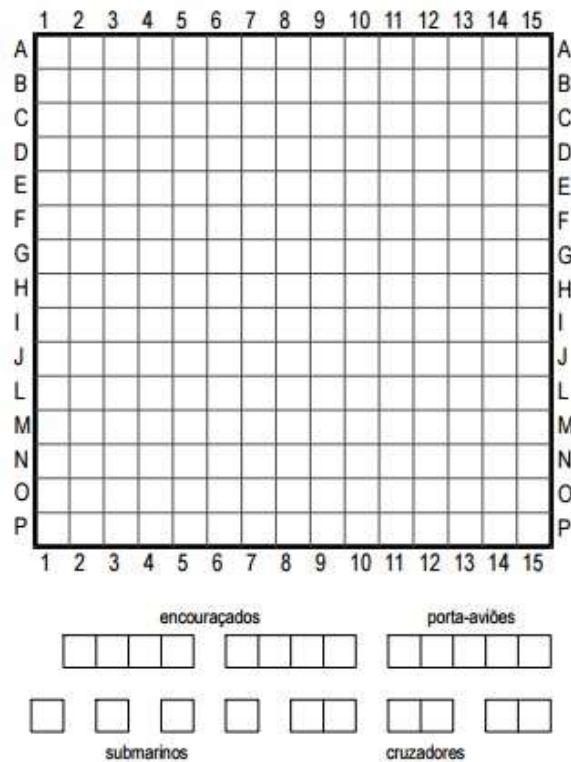


Figura 1 – Tabuleiro do Jogo e Peças Suportadas

Para este jogo a ser desenvolvido, serão consideradas as seguintes peças que poderão ser posicionadas na **horizontal** (deslocamento sempre à direita) ou **vertical** (deslocamento sempre abaixo):

- Código 1 - (encouraçados – 4 posições);
- Código 2 - (porta-aviões – 5 posições);
- Código 3 - (submarinos – 1 posição);
- Código 4 - (cruzadores – 2 posições);

O jogo deverá atender aos seguintes requisitos:

1. O sistema deverá receber dois arquivos de entrada para processamento do programa, denominados: **jogador1.txt (ID=J1)** e **jogador2.txt (ID=J2)**; e deverá gerar um único arquivo de saída com o nome **resultado.txt**;
2. O arquivo deverá ter o seguinte formato, simulando o posicionamento das peças no tabuleiro (ver Figura abaixo):

1;A2V|C7H

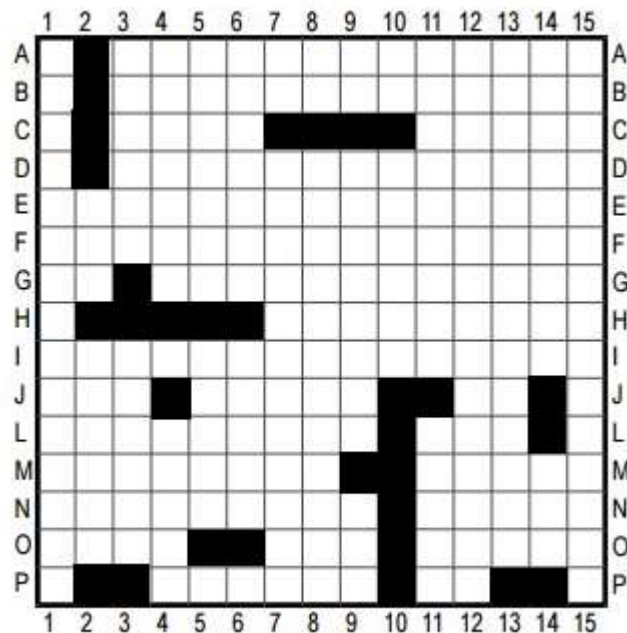
2;H3H|L10V
 3;O5|O6|M9|J4|G3
 4;J10H|J14V|P13H|P2H
 # Jogada
 T;A1|A2|A3|A4|B5|O6|O7|J8|P9|D10|G11|P12|P14|N15|M16|D14|A5|B11|C12

Sendo:

1;A2V|C7H → <código da peça>;<posição da peça>;<direcionamento da peça>

Jogada → <delimitador do grupo de instruções de posicionamento das peças e do grupo de instruções de lançamento de torpedo>

T;A1|A2|A3 → <código de torpedo>;<posições onde os torpedos serão disparados>



3. As peças de código 3 (três) serão as únicas a não possuírem direcionamento, por ocuparem apenas uma posição no tabuleiro;
4. Cada jogador terá direito a:
 - Posicionar 2 peças de código 1 e 2 peças de código 2;
 - Posicionar 5 peças de código 3;
 - Posicionar 4 peças de código 4; e
 - Disparar exatamente 20 tiros de torpedo.

As jogadas e tiros deverão ser validados de acordo com a quantidade informada e caso alguma esteja fora da quantidade **exata**, a saída a ser escrita no arquivo será deverá ser `ERROR_VALIDATION`, finalizando o programa em seguida;

5. O sistema deverá ler os dois arquivos de entrada, criando duas coleções de dados que representem as peças posicionadas no tabuleiro.

6. As peças a serem posicionadas no tabuleiro não devem se sobrepor às peças do próprio jogador, para isso considere:
- 1;A2H (A primeira peça ocupará as posições A2 + A3 + A4 + A5)
- 4;A5H (A segunda peça ocupará as posições A5 + A6)
- Ou seja, ambas as peças ocupam a mesma posição A5. Dessa forma, deverá ser escrita no arquivo de saída a mensagem: `ERROR_INVALID_INPUT_FILE`. A mensagem só deverá ser apresentada após a validação dos dois arquivos de entrada, e caso algum deles esteja incorreto, o jogo deve ser finalizado antes de executar as jogadas;
7. Os torpedos a serem disparados devem representar posições existentes dentro das dimensões do tabuleiro. Caso alguma das posições dos torpedos esteja fora dos limites do tabuleiro, a seguinte saída deve ser escrita no arquivo de saída e o jogo finalizado: `ERROR_INVALID_COORDINATE`;
8. A linha que representa os torpedos a serem disparados deverá ser lida e para cada posição que representa o ponto onde o torpedo será direcionado, o sistema deverá analisar se o adversário possui algum navio na posição. Se o alvo na posição for destruído parcialmente, deverá ser somado ao jogar **3 pontos**. Caso o alvo seja destruído integralmente, deverá ser somado ao jogador **5 pontos**. Caso nenhum alvo seja acertado, nada deve ser computado.
9. Caso o processamento ocorra com sucesso, o aplicativo deverá gerar a seguinte saída no arquivo **resultado.txt**:
- **quem** foi o jogador ganhador;
 - **quantos** alvos foram acertados pelo jogador ganhador (parciais e totais);
 - **quantos** alvos restaram no tabuleiro do jogador adversário (parciais e totais); e
 - **qual** foi a pontuação final do jogador ganhador.
 -

Formato da escrita no arquivo (separador SPACE → " "):

```
J1 12P 3T 5P 6T 34
ID_GANHADOR ACERTOS_PARCIAIS ACERTOS_TOTAIS REST_PARCIAIS REST_TOTAIS PONTUACAO
```

10. Caso o processamento ocorra com sucesso e o resultado seja **EMPATE**, utilize o formato descrito no item 10 para escrever o resultado dos 2 (dois) jogadores no arquivo de saída (resultado.txt), sendo o resultado de **J1** na primeira linha e o resultado de **J2** na segunda linha.