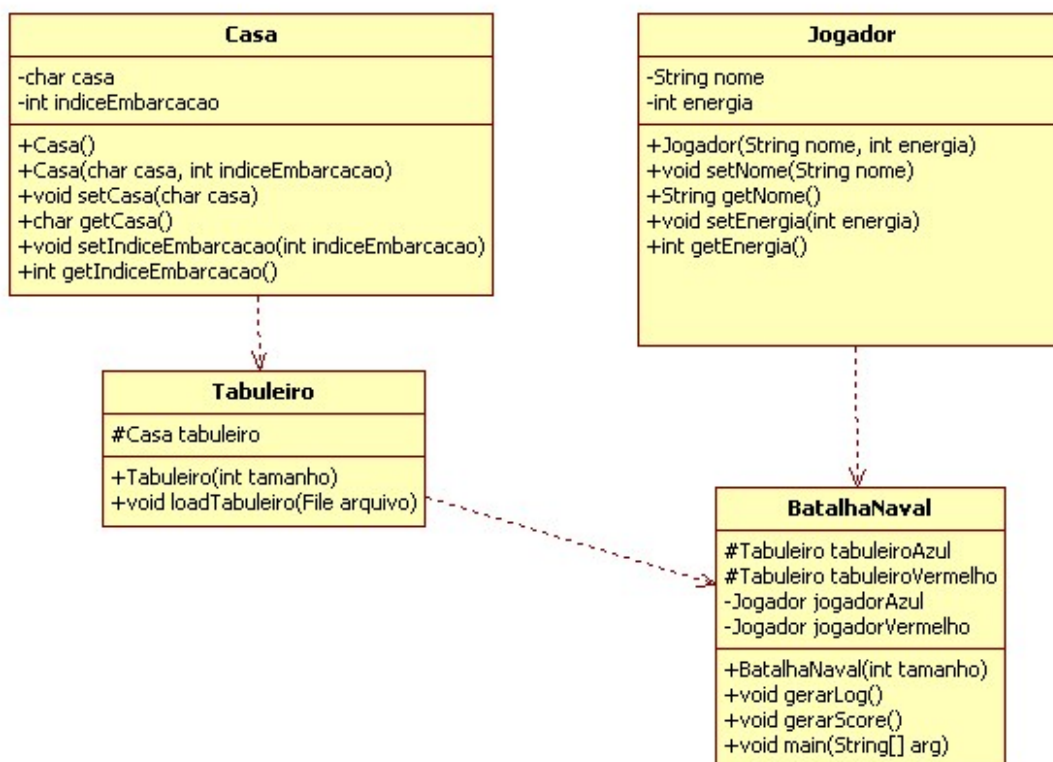


TRABALHO DE PROGRAMAÇÃO ORIENTADA A OBJETO BATALHA NAVAL

INTRODUÇÃO

Este trabalho consiste em desenvolver uma variação orientada a objeto do conhecido jogo batalha naval, considerando os tabuleiros como sendo uma matriz 10x10 e as seguintes embarcações: **submarino (S)**, **destroyer (DD)**, **cruzador (CCCC)** e **porta-aviões (PPPPP)**.

Considere o diagrama de classe abaixo para implementação do jogo.



O programa deve carregar as embarcações azuis e vermelhas nos tabuleiros através de arquivos textos que possuem as posições das embarcações, onde os jogadores devem acertá-las dando tiros (informando a linha e a coluna do tabuleiro e artilharia que vai usar (torpedo ou canhão).

O objetivo dessa versão do jogo é afundar todas as embarcações do inimigo disposta no tabuleiro.

DESCRIÇÃO DO JOGO

O programa deve solicitar o nome do jogador azul e o um **arquivo texto**, que tem as posições de **1 (um) porta-aviões azul**, **2 (dois) cruzador azul**, **3 (três) destroyers azul**, **5 (cinco) submarinos azul**. Em seguida solicita o nome do jogador vermelho e o seu arquivo com **1 (um) porta-aviões vermelho**, **2 (dois) cruzador vermelho**, **3 (três) destroyers vermelho**, **5 (cinco) submarinos vermelho**.

Antes de iniciar o jogo cada jogador recebe uma **energia (life) igual a 40 (quarenta)**, esse atributo serve para medir a energia de cada jogador, para tornar o jogo emocionante e competitivo, visto que para **cada tiro de canhão o jogador perde 2 (dois) níveis de energia** e para **um tiro de torpedo ele perde 4 (quatro) níveis**.

Caso uma embarcação seja atingida por um torpedo, deve ser destruída mais parte da embarcação ao redor da casa alvo, como mostra o exemplos abaixo.

Exemplo 1: Considere o tabuleiro abaixo, e que o jogador deseja disparar um torpedo na casa [4,3].

	C	C	C	C

Por ser um torpedo, as casas [4,2] e [4,4] também serão destruídas, como mostra o tabuleiro abaixo

	X	X	X	C

Exemplo 2: Considere o tabuleiro abaixo, e que o jogador deseja disparar um torpedo na casa [4,2].

				C
			C	
		C		
	C			

Por ser um torpedo, e só existir uma parte da embarcação no raio de uma casa ao redor da casa alvo, somente a casas [3,3] será destruída, como mostra o tabuleiro abaixo

				C
			C	
		X		
	X			

Exemplo 3: Considere o tabuleiro abaixo, e que o jogador deseja disparar um torpedo na casa [5,1].

				C
			C	
		C		
	C			

Nesse caso nenhuma parte da embarcação será destruída, como pode ser vista no tabuleiro abaixo.

				C
			C	
		C		
	C			
X				

Para os tiros de canhão só deve ser destruída a casa alvo.

Cada jogador deve receber na inicialização do jogo o mesmo número de artilharia, seguindo a seguinte ordem: **4 (quatro) torpedos e 12 (doze) bala de canhão.**

Para aumentar a competitividade do jogo deve ser implementado o seguinte: **para cada tiro bem sucedido de torpedo o jogador ganha 2 (dois) níveis e 1 (um) nível para cada tiro de canhão bem sucedido.**

Quando o tiro for disparado após o jogador informar as coordenadas (linhas e colunas) e a artilharia, o programa deve imprimir as seguintes mensagens: **Tipo invalido** (se o tiro cair fora dos limites do tabuleiro), **Água** (se o tiro acertou a água), o **Tipo da embarcação** (se acertou alguma embarcação), **Afundou** (se todas as casas da embarcação foram atingidas) e **Local já atingido** (se o tiro acertar uma casa já atingida).

DESCRIÇÃO DO PROGRAMA

O programa consiste de 3 (três) etapas:

ETAPA 1: CARREGANDO OS TABULEIROS

Dispor todas as embarcações nos tabuleiros (azul e vermelho) a partir dos arquivos textos, dos jogadores (azul e vermelho). Sendo que nenhuma embarcação deve se encostar (nem mesmo na diagonal).

O arquivo texto tem a seguinte estrutura:

<linha> <coluna> <embarcação> <índice>

Onde **<embarcação>** seria o tipo das embarcações, que pode ser do seguinte tipo: **S (submarino)**, **D (destroyer)**, **P (porta-aviões)** e **C (cruzador)**. E a **tag <índice>** será o índice da embarcação no **vetor embarcações**, o índice de embarcações vai de **0 (zero)** a **10 (dez)**.

Exemplo de um arquivo texto:

```
0    1    D    0
0    2    D    0
2    3    S    1
2    1    D    2
3    1    D    2
```

Abaixo mostramos um exemplo do tabuleiro gerado pelo arquivo acima:

	D	D	
	D		S
	D		

Essa etapa carrega os tabuleiros do jogo com água e com as embarcações presentes nos arquivos carregados pelos usuários. Considerando **0 (zero) como água**, e as embarcações são identificadas com as letras **S (submarino)**, **D (destroyer)**, **P (porta-aviões)** e **C (cruzador)**.

O construtor **Tabuleiro(int tamanho)** define o tamanho do tabuleiro e preenche o tabuleiro com água.

0	D	D	0
0	0	0	0
0	D	0	S
0	D	0	0

Para carregar as embarcações no tabuleiro, desenvolva um método chamado **loadTabuleiro(File arquivo)** esse método retorna um valor **booleano** para informar se foi possível carregar o tabuleiro (se o tabuleiro é válido).

Para o tabuleiro ser válido o programa deve verificar se as embarcações: não ultrapassam os limites do tabuleiro, uma embarcação não ocupa uma casa já ocupada e que não existem embarcações encostas.

Os tabuleiros devem ser impressos a cada rodada de disparos, no início os tabuleiros devem ser impressos com # em cada posição, como é mostrado abaixo:

Tabuleiro Azul**Energia:** 40**Canhão:** 12 **Torpedo:** 4

```

0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
  0 1 2 3 4 5 6 7 8 9

```

Tabuleiro Vermelho**Energia:** 40**Canhão:** 12 **Torpedo:** 4

```

0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
  0 1 2 3 4 5 6 7 8 9

```

Disparo Jogador Azul

Linha (0-9): 2

Coluna (0-9): 4

Artilharia (c: Canhão e t: torpedo): c

Após um disparo, o programa deve limpar a tela e imprimir os novos tabuleiros com um **X** na casa atingida caso acerte a água e com **S**, **D**, **C** ou **P** caso acerte um **submarino**, **destroyer**, **cruzador** ou **porta-aviões** respectivamente. Como mostramos nos exemplos abaixo:

Tabuleiro Azul**Energia:** 38**Canhão:** 11 **Torpedo:** 4

```

0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
  0 1 2 3 4 5 6 7 8 9

```

Tabuleiro Vermelho**Energia:** 40**Canhão:** 12 **Torpedo:** 4

```

0 #####
1 #####
2 ##### X #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
  0 1 2 3 4 5 6 7 8 9

```

O jogador azul acertou a água.

Disparo Jogador Vermelho

Linha (0-9): 8

Coluna (0-9): 3

Artilharia (c: Canhão e t: torpedo): t

Tabuleiro Azul**Tabuleiro Vermelho**

Energia: 38

Canhão: 11 Torpedo: 4

Energia: 38

Canhão: 12 Torpedo: 3

0 # # # # # # # # #

1 # # # # # # # # #

2 # # # # # # # # #

3 # # # # # # # # #

4 # # # # # # # # #

5 # # # # # # # # #

6 # # # # # # # # #

7 # # # # # # # # #

8 # # # **D D** # # # # #

9 # # # # # # # # #

0 1 2 3 4 5 6 7 8 9

0 # # # # # # # # #

1 # # # # # # # # #

2 # # # # **X** # # # # #

3 # # # # # # # # #

4 # # # # # # # # #

5 # # # # # # # # #

6 # # # # # # # # #

7 # # # # # # # # #

8 # # # # # # # # #

9 # # # # # # # # #

0 1 2 3 4 5 6 7 8 9

O jogador vermelho acertou e afundou um *destroyer*.

Disparo Jogador Azul

Linha (0-9):

ETAPA 2: BATALHA

A batalha inicia com a leitura das coordenadas dos disparos (linha e coluna) e a artilharia (canhão ou torpedo), para gerar alguma mensagem conveniente.

Os disparos devem ser lidos do teclado, ou seja, os jogadores digitam as coordenadas de cada disparo.

O jogo é finalizado quando todas as embarcações de um jogador sejam afundadas, ou um dos jogadores perde toda a energia ou quando um dos jogadores não tenha nenhuma munição para disparar.

Para essa etapa, considere um vetor de embarcações de tamanho 11 (onze), onde cada índice representa uma embarcação. Este vetor servirá para indicar quantas casas de uma embarcação faltam ainda para serem atingidas.

Use contadores para os disparos dados pelos jogadores, com canhão e torpedo, esses contadores servem, por exemplo, para organizar uma competição de batalha, vendo quem afunda todas as embarcações com o menor disparo, levando em consideração disparos repetidos.

A cada disparo dado pelos jogadores, deve chamar o método `gerarLog()`, que grava em um arquivo texto (*log.txt*) os jogadores, as coordenadas dos disparos e as mensagens que corresponde a cada disparo efetuado. Essas mensagens podem ser: **Disparo inválido** (se o tiro caiu fora do tabuleiro), **Local já atingido** (se o tiro acerta uma casa do tabuleiro já atingida), **Água** (se o tiro acerta a água), **Acertou um embarcação** (se acerta uma embarcação) e **Acertou e afundou uma embarcação** (se afundar uma embarcação).

Marque com um **X** a casa do tabuleiro atingida pelo disparo. Caso atinja uma embarcação, decremente de **1 (um)** a posição do vetor embarcações que representa a embarcação atingida. Se o valor da posição do vetor ficar **nulo (zero)**, imprima a mensagem **Acertou e afundou uma embarcação**.

Abaixo mostramos um exemplo do arquivo *log.txt*:

Jogador azul (Paulo) [0,2]: Água
Jogador vermelho (Alberto) [1,3]: Acertou e afundou um submarino
Jogador azul (Paulo) [2,5] Acertou um *destroyer*
...

ETAPA 3: ESTATÍSTICA

No final do jogo apresente os dados estatísticos da batalha: percentuais de disparos na água, certos não repetidos, certos repetidos e inválidos.

Para essa etapa, considere os contadores de disparos dados pelos jogadores, com canhão e torpedo.

Ao final do jogo o programa deve chamar o método `gerarScore()`, que grava em um arquivo texto (*score.txt*) os jogadores, e os dados estatísticos da batalha.

Abaixo mostramos um exemplo do arquivo *score.txt*:

Jogador azul (Paulo):
10 tiros de canhão: 50% água, 36% certos, 12% repetidos e 2% inválidos
4 tiros de torpedo: 50% água, 25% certos, 0% repetidos e 25% inválidos
Jogador vermelho (Alberto):
8 tiros de canhão: 10% água, 88% certos, 0% repetidos e 2% inválidos
4 tiros de torpedo: 25% água, 75% certos, 0% repetidos e 0% inválidos