

Exercício 6: Batalha naval (4,0) – Avaliação 26/04/2024.

Desenvolva um programa em java que simule uma batalha naval. Implemente as seguintes classes:

1. **MAPA:** Representa o oceano através de uma matriz de caracteres. O tamanho deve ser grande o suficiente para comportar as embarcações e os espaços sem embarcações. O mapa vai ser mostrado aos jogadores inicialmente com o valor 'O' (oculto). Os espaços vazios (V) devem ser pelo menos 70% de todo espaço disponível.
2. **Embarcação:** Representa uma embarcação a ser colocada no mapa. Devem ser criados os seguintes tipos de embarcação por herança: (i) – 2 Porta Aviones, representados pela letra (a), ocupando 8 posições no mapa, (ii)- 3 destroyers, representados pela letra (d), ocupando 5 posições no mapa, (iii) – 4 submarinos, representados pela letra (s), ocupando 4 posições no mapa, (iv) 5 fragatas, representados pela letra (f), ocupando 3 posições no mapa e finalmente (v) 6 botes, representados pela letra (b), ocupando 2 posições no mapa. As embarcações deve ser alocadas no mapa de forma horizontal e vertical de maneira proporcional. Não podem ter embarcações sem um espaço vazio adjacente no mapa.
3. **Jogador:** Representa um jogador, sendo necessário calcular a **pontuação** do jogador em cada acerto a uma embarcação
4. **Jogo:** Controla toda a lógica do jogo. O acerto é sempre inferido através de 2 coordenadas na matriz do mapa. Quando um jogador acerta uma embarcação, deve ser marcado um X no mapa (ao acertar um pedaço da embarcação o jogador neutraliza aquela embarcação). Caso alguma embarcação seja acertada pelo adversário ela deve ser marcada com Y no mapa. É importante que o mapa parcial seja exibido a ambos jogadores em cada rodada, sendo assim, a classe jogo fica responsável por atualizar o estado do mapa para os jogadores a cada rodada. Finalmente, o jogo deve informar qual jogador ganhou, o ganhador é o jogador que abateu 5 ou mais embarcações. Deve ser mantido um log de quais embarcações o jogador abateu, que deve ser exibido para os jogadores no final do jogo. Este log também deve ser persistido em texto, para cada partida. Preferencialmente, o programa deve utilizar sockets, com o controle do jogo no servidor

Bônus 1: Uso de sockets (+0,5).

Bônus 2: Implementação de jogo em duplas (+0,5).