

Trabalho em Java:

Implementação do Jogo Batalha Naval

Objetivo

Desenvolver uma aplicação em Java que simule o jogo Batalha Naval, jogado via terminal. A aplicação deverá permitir que o jogador ataque coordenadas no tabuleiro e receba feedback sobre os ataques, incluindo se acertou ou errou navios.

Descrição

Você deverá implementar o jogo Batalha Naval em um tabuleiro 8x8 que contém 10 navios posicionados aleatoriamente. O jogador poderá atacar posições no tabuleiro até que todos os navios sejam destruídos ou tenha completado 30 tentativas. O jogo deve ser jogado via terminal e deve fornecer feedback ao jogador após cada ataque.

Requisitos

1. Estrutura do Projeto:

- Utilize a linguagem Java somente com o conteúdo aprendido em sala de aula (unidade 1 até 6, matrizes e classe Random). Os únicos *import* devem ser: `import java.util.Scanner;` e `import java.util.Random;`
- Todo o código deve ser implementado em uma única classe.
- O código deve ser bem documentado e seguir boas práticas de programação.

2. Inicialização do Tabuleiro:

- Crie um tabuleiro 8x8 representado por uma matriz bidimensional de caracteres.
- Preencha o tabuleiro com água ('~') inicialmente.

Figura 1 - Exemplo de inicialização

	0	1	2	3	4
0	~	~	~	~	~
1	~	~	~	~	~
2	~	~	~	~	~
3	~	~	~	~	~
4	~	~	~	~	~

3. Posicionamento dos Navios:

- Posicione aleatoriamente dez navios ('N') no tabuleiro utilizando a classe Random do Java.
- Certifique-se de que os navios não se sobreponham.
- A posição dos navios não deve ser revelada para o jogador

4. Interação com o Jogador:

- Peça ao jogador para inserir coordenadas de ataque (linha e coluna).

- Valide as coordenadas inseridas (devem estar dentro dos limites do tabuleiro).

Figura 2 - Exemplo de jogada com feedback

```
Digite as coordenadas para atacar (linha e coluna, separadas por espaço):
1 2
Você errou.
  0 1 2 3 4
0 ~ ~ ~ ~ ~
1 ~ ~ 0 ~ ~
2 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~
```

5. Feedback dos Ataques:

- Informe ao jogador se o ataque acertou um navio ou errou.
- Atualize o tabuleiro com 'A' para acertos e 'X' para erros.
- Mostre o tabuleiro ao jogador após cada ataque, mas sem revelar as posições dos navios restantes.
- Mostre ao jogador se é uma casa que ele já atacou.
- Mostre ao jogador se ele jogou em uma posição inválida.

6. Condição de Parada:

- O jogo termina quando todos os navios forem destruídos, nesse caso ele vence ou o jogador tenha completado um total de 30 jogadas e não tenha encontrado todos os navios (nesse caso ele perde).
- Informe ao jogador se ele venceu ou foi derrotado e revele todas as posições dos navios.

Entregáveis

1. Código Fonte:

- Todo o código fonte do projeto no arquivo Java, junto com a documentação.
- O código deve ser entregue via AVA.

2. Documento:

- Descrição das funcionalidades implementadas.

3. Imagens:

- Capturas de telas demonstrando a funcionalidade do jogo. As imagens devem ser enviadas todas juntas em um documento .PDF com a descrição de cada uma delas.

Critérios de Avaliação

- Correção: o programa deve funcionar corretamente e seguir as regras do jogo.
- Qualidade do Código: o código deve ser bem estruturado, legível e seguir boas práticas de programação.
- Interação com o Usuário: a interação deve ser intuitiva, com mensagens de erro adequadas quando necessário.
- Documentação: A descrição das funcionalidades deve ser completa.

- **Imagens:** As imagens devem representar significativamente as funcionalidades do jogo.

Prazos

- **Entrega do Código Fonte e Documentação:** 02/12/24
- **Apresentação do Projeto – no máximo:** 02/12/24

Atenção: apresentar o trabalho o código fonte e documentação devem estar postados no AVA.

Instruções Adicionais

- Trabalhe individualmente ou em grupos de até 2 pessoas
- Qualquer dúvida ou problema deve ser comunicada ao professor ou monitor da disciplina o mais breve possível.

Bons códigos! 😊