Alunos: Guilherme Da Silva Geisler

Descrição das Funcionalidades do Jogo Batalha Naval

1. Estrutura do Projeto

- **Linguagem Utilizada:** Java, utilizando o conteúdo aprendido nas unidades 1 a 6 (matrizes e classe Random).
- **Classe Principal:** Toda a lógica do jogo está implementada na classe BatalhaNaval.
- **Boas Práticas:** Código bem documentado e estruturado, seguindo as boas práticas de programação 😉.

2. Inicialização do Tabuleiro

- **Método:** carregarTabuleiro()
- **Descrição:** Inicializa o tabuleiro 8x8 com o caractere '~' representando água.
- **Código:**

```
private static void carregarTabuleiro() {
       for (int i = 0; i < TAMANHO_TABULEIRO; i++) {
           for (int j = 0; j < TAMANHO_TABULEIRO; j++) {
               tabuleiro[i][j] = AGUA;
           }
       }
   }
```

3. Posicionamento dos Navios

- **Método:** distribuirNavios()
- **Descrição:** Posiciona aleatoriamente 10 navios no tabuleiro, utilizando a classe Random do Java. Certifica-se de que os navios não se sobreponham e que a posição dos navios não seja revelada para o jogador.
- **Código:**

```java
private static void distribuirNavios() {
    int naviosDistribuidos = 0;

    while (naviosDistribuidos < QUANTIDADE_NAVIOS) {
        int linha = random.nextInt(TAMANHO_TABULEIRO);
        int coluna = random.nextInt(TAMANHO_TABULEIRO);

        if (!navios[linha][coluna]) {
            navios[linha][coluna] = true;
            naviosDistribuidos++;
        }
    }
}
```

4. Interação com o Jogador

- **Método:** realizarJogada()
- **Descrição:** Solicita ao jogador as coordenadas de ataque (linha e coluna) e valida as coordenadas inseridas. O jogador é informado se o ataque acertou um navio ou errou.
- **Código:**

```
private static void realizarJogada() {
    System.out.println("\nRodada #" + (tentativas + 1));
    boolean entradaValida = false;
    int linha = -1;
    int coluna = -1;
    while (!entradaValida) {
        System.out.print(
"Digite as coordenadas para atacar (linha e coluna, separadas por espaço): ");
        String entrada = sc.nextLine();
        String[] tokens = entrada.trim().split("\\s+");
        if (tokens.length == 2) {
            boolean linhaValida = tokens[0].matches("\\d+");
            boolean colunaValida = tokens[1].matches("\\d+");
            if (linhaValida && colunaValida) {
                linha = Integer.parseInt(tokens[0]);
                coluna = Integer.parseInt(tokens[1]);
                // Valida as coordenadas inseridas pelo jogador
                if (linha >= 0 && linha < TAMANHO_TABULEIRO && coluna >= 0 && coluna <
 TAMANHO_TABULEIRO &&
                    tabuleiro[linha][coluna] != TENTATIVA_AGUA && tabuleiro[linha][coluna] !=
 TENTATIVA_ACERTO) {
                    entradaValida = true;
                } else {
                    System.out.println("Coordenadas inválidas ou já atacadas! Tente novamente.");
                }
            } else {
                System.out.println("Coordenada inválida! Por favor, insira números inteiros.");
            }
        } else {
            System.out.println(
"Coordenada inválida! Por favor, insira duas coordenadas (linha e coluna).");
        }
    }
    // Verifica se o ataque acertou um navio
    boolean acertou = navios[linha][coluna];
    tabuleiro[linha][coluna] = acertou ? TENTATIVA_ACERTO : TENTATIVA_AGUA;
    // Atualiza o estado do jogo
    if (acertou) {
        naviosRestantes--;
        System.out.println("Você acertou um navio!");
    } else {
        System.out.println("Você errou o ataque.");
    }
    tentativas++;
}
```

5. Feedback dos Ataques

- **Método:** exibirTabuleiro()
- **Descrição:** Mostra o tabuleiro ao jogador após cada ataque, sem revelar as posições dos navios restantes. Informa ao jogador se ele já atacou uma determinada posição ou se inseriu uma posição inválida.
- **Código:**

```
private static void exibirTabuleiro() {
    System.out.println("\n  0 1 2 3 4 5 6 7");
    for (int i = 0; i < TAMANHO_TABULEIRO; i++) {
        System.out.print(i + " ");
        for (int j = 0; j < TAMANHO_TABULEIRO; j++) {
            char conteudo = tabuleiro[i][j];
            if (conteudo == NAVIO && naviosRestantes > 0) {
                conteudo = AGUA;
            }
            System.out.print(conteudo + " ");
        }
        System.out.println();
    }
}
```

6. Condição de Parada

- **Método:** exibirResultadoFinal()
- **Descrição:** O jogo termina quando todos os navios são destruídos ou quando o jogador atinge 30 tentativas. O jogador é informado se venceu ou foi derrotado e todas as posições dos navios são reveladas.
- **Código:**

```
    private static void exibirResultadoFinal() {
        System.out.println("\nTabuleiro final:");
        exibirTabuleiroFinal();

        if (naviosRestantes == 0) {
            System.out.println(
"Parabéns! Você destruiu todos os navios!");
        } else {
            System.out.println(
"Fim de jogo! Você não conseguiu destruir todos os navios.");
        }
    }

    private static void exibirTabuleiroFinal() {
        System.out.println("\n  0 1 2 3 4 5 6 7");
        for (int i = 0; i < TAMANHO_TABULEIRO; i++) {
            System.out.print(i + " ");
            for (int j = 0; j < TAMANHO_TABULEIRO; j++) {
                char conteudo = navios[i][j] ? NAVIO :
 tabuleiro[i][j];
                System.out.print(conteudo + " ");
            }
            System.out.println();
        }
    }
}
```

Impressões de Saída:

Tela Inicial
Descrição: Exibe o tabuleiro inicial com água ('~') em todas as posições.

Impressão:

```
  0 1 2 3 4 5 6 7
0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
```

Rodada #1

Print:

```
  0 1 2 3 4 5 6 7
0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~

Rodada #1
Digite as coordenadas para atacar (linha e coluna, separadas por espaço):
```

Solicitação de Coordenadas de Ataque

Descrição: Solicita ao jogador que insira as coordenadas para atacar (linha e coluna, separadas por espaço).

Impressão:

Digite as coordenadas para atacar (linha e coluna, separadas por espaço): 1 1

Feedback de Ataque

Descrição: Mostra o tabuleiro atualizado após cada ataque, indicando se foi um acerto ('X') ou um erro ('O').

Impressão:

Você errou o ataque.

0 1 2 3 4 5 6 7
0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ **O** ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~

Rodada #2

Print:

```
Digite as coordenadas para atacar (linha e coluna, separadas por espaço): 1 1
Você errou o ataque.

  0 1 2 3 4 5 6 7
0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ O ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~

Rodada #2
```

Condição de Vitória

Descrição: Exibe a mensagem de vitória quando todos os navios são destruídos.

Impressão:

Tabuleiro final:

```
  0 1 2 3 4 5 6 7
0 ~ ~ N ~ ~ N ~ N
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ N ~ ~ ~ ~ ~
3 ~ ~ ~ ~ N ~ ~ ~
4 N ~ ~ ~ ~ ~ ~ ~
5 ~ N ~ ~ N ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ N ~ N ~ ~ ~
```

Parabéns! Você destruiu todos os navios!

Condição de Derrota

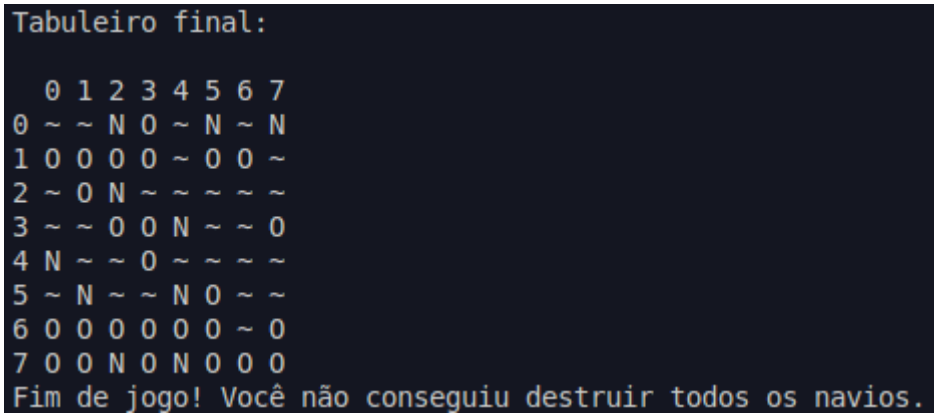
Descrição: Exibe a mensagem de derrota quando o jogador atinge o limite de 30 tentativas sem destruir todos os navios.

Impressão:

Tabuleiro final:

```
  0 1 2 3 4 5 6 7
0 ~ ~ N O ~ N ~ N
1 O O O O ~ O O ~
2 ~ O N ~ ~ ~ ~ ~
3 ~ ~ O O N ~ ~ O
4 N ~ ~ O ~ ~ ~ ~
5 ~ N ~ ~ N O ~ ~
6 O O O O O O ~ O
7 O O N O N O O O
Fim de jogo! Você não conseguiu destruir todos os navios.
```

Print:


Tabuleiro final:
0 1 2 3 4 5 6 7
0 ~ ~ N O ~ N ~ N
1 O O O O ~ O O ~
2 ~ O N ~ ~ ~ ~ ~
3 ~ ~ O O N ~ ~ O
4 N ~ ~ O ~ ~ ~ ~
5 ~ N ~ ~ N O ~ ~
6 O O O O O O ~ O
7 O O N O N O O O
Fim de jogo! Você não conseguiu destruir todos os navios.