

Trabalho em Java: Implementação do Jogo Batalha Naval

FURB BCC - Ciências da Computação, 1 Semestre
Nomes: Gustavo Baum, Guilherme Bez.

Documentação das Funcionalidades:

Método que dá a opção de se recomeçar o jogo logo após concluí-lo:

```
//Inicio comando de 'jogar novamente':
```

```
String Recomeçar;  
do {
```

```
System.out.println(x:"\nDeseja jogar novamente? (s) - Sim, (n) - Não");  
Recomeçar = entrada.nextLine();  
} while (Recomeçar == "S" || Recomeçar == "s");  
  
entrada.close();
```

Método que popula as matrizes do tabuleiro e das coordenadas de base:

```
//Inicializador do tabuleiro:  
  
char[][] Mapa = new char[8][8];  
char[][] Navios = new char[8][8];  
int [] Ordenação = new int[9];  
int[] Coordenadas = new int[30];  
  
System.out.println(x:"Escreva qualquer coisa para começar: ");  
String Iniciar = entrada.nextLine();  
  
System.out.println(x:"\nIniciando jogo...\n");  
  
//Populando posições(Ordenação):  
  
for (int i = 0; i < Ordenação.length; i++) {  
    Ordenação[i] = i;  
    System.out.print(Ordenação[i] + " ");  
}  
System.out.println();  
  
//Popular vetores do Mapa:  
  
for (int Linha = 0; Linha < Mapa.length; Linha++) {  
    for (int Coluna = 0; Coluna < Mapa[Linha].length; Coluna++) {  
        Mapa[Linha][Coluna] = '~';  
    }  
}  
  
for (int i = 0; i < Mapa.length; i++) {  
    System.out.print(Ordenação[i + 1] + " ");  
    for (int j = 0; j < Mapa.length; j++) {  
        System.out.print(Mapa[i][j] + " ");  
    }  
    System.out.println();  
}
```

Método que randomiza a posição dos navios no tabuleiro:

```
//Metodo que randomiza as posições dos Navios no Mapa:  
  
randomizador(Mapa, Navios, Linha:0, Coluna:0, random);
```

```
//Metodo para a randomização dos Navios:  
  
private void randomizador(char[][] Mapa, char[][] Navios, int Linha, int Coluna, Random random) {  
    int Verificador = 10;  
    for (int n = 0; n < Verificador; n += 0) {  
        Linha = random.nextInt(bound:8);  
        Coluna = random.nextInt(bound:8);  
        if (Navios[Linha][Coluna] != 'N') {  
            Navios[Linha][Coluna] = 'N';  
            Verificador--;  
        }  
    }  
    System.out.println(x:"\nNavios posicionados\n");  
}
```

Método que realiza a adquiere a coordenada indicada pelo jogador e seus respectivos resultados:

```
//Iniciando o jogo...  
  
int Linha = 0;  
int Coluna = 0;  
int acertos = 0;  
int Tentativas = 0;  
do {  
    Tentativas++; //Computando Tentativas;  
  
    //Parte de interação do usuário usando um método de base:  
  
    jogar(entrada, Mapa, Navios, Coordenadas, Linha, Coluna, Tentativas, acertos);  
  
    System.out.println();  
  
    for (int i = 0; i < Ordenação.length; i++) {  
        System.out.print(Ordenação[i] + " ");  
    }  
  
    System.out.println();  
  
    for (int i = 0; i < Mapa.length; i++) {  
        System.out.print(Ordenação[i + 1] + " ");  
        for (int j = 0; j < Mapa.length; j++) {  
            System.out.print(Mapa[i][j] + " ");  
        }  
        System.out.println();  
    }  
}  
} while (Tentativas != 30 && acertos != 10);  
if (acertos == 10) {  
    System.out.println(x:"\nVOCÊ VENCEU\nTODA A FROTA INIMIGA FOI DESTRUÍDA\n");  
} else {  
    System.out.println(x:"\nVOCÊ PERDEU\nAPESAR DOS SEUS EXFORÇOS, A FROTA INIMIGA CONSEGUIU FUGIR\n");  
  
    for (int i = 0; i < Ordenação.length; i++) {  
        System.out.print(Ordenação[i] + " ");  
    }  
  
    System.out.println();  
  
    for (int i = 0; i < Mapa.length; i++) {  
        System.out.print(Ordenação[i + 1] + " ");  
        for (int j = 0; j < Navios.length; j++) {  
            if (Navios[i][j] != 'N') {  
                System.out.print("~" + " ");  
            } else {  
                System.out.print(Navios[i][j] + " ");  
            }  
        }  
        System.out.println();  
    }  
}
```

Método jogar (OBS: foram inseridos métodos que visam otimizar a experiência do jogador com erros, como a reinicialização da obtenção das coordenadas se, e somente se, as coordenadas fornecidas não forem reais, como no caso de se extrapolar o limite do tabuleiro, de 11 à 88, e em relação ao fornecimento de letras erroneamente):

```
//Método que contele a escolha das coordenadas à atacar e seus respectivos resultados:
private int jogar(Scanner entrada, char[][] Mapa, char[][] Navios, int[] Coordenadas, int Linha, int Coluna, int Tentativas, int acertos) {
    int Verificador = 0;
    do {
        try {
            System.out.println(x:"\nDigite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2): ");

            String resposta = entrada.nextLine().replace(target: " ", replacement:"");
            int a = Integer.parseInt(resposta);

            //Se a coordenada for menor que 11, será requisitado uma nova coordenada, pois coordenada menor que 11 não existe no jogo;

            while (a < 11) {
                System.out.println(x:"\nCoordenada inválida, por favor tente novamente ");
                resposta = entrada.nextLine().replace(target: " ", replacement:"");
                a = Integer.parseInt(resposta);
            }

            Verificador = 0;

            for (int i = 0; i < Tentativas; i++) {
                if (Coordenadas[i] == a) {
                    Verificador = 1;
                }
            }
            Coordenadas[Tentativas] = a;

            if (Verificador != 1) {
                for (int i = 0; i < 9; i++) {
                    for (int j = 1; j < 9; j++) {
                        String x = i + "" + j;
                        int b = Integer.parseInt(x);

                        if (a == b) {
                            Linha = i;
                            Coluna = j;
                        }
                    }
                }
            }
        } catch {
            //Verificação do mapa:

            if (Verificador == 0) {
                Linha--;
                Coluna--;
            }

            //Se o navio estiver na coordenada indicada pelo jogador, o mapa marcará 'X' na posição:

            if (Navios[Linha][Coluna] == 'N') {
                Mapa[Linha][Coluna] = 'X';
                acertos++;
                System.out.println("\nVoce acertou o navio. Faltam: " + (10 - acertos) + ".");

                //Se o navio NÃO estiver na coordenada indicada pelo jogador, o mapa marcará "0" na posição:

            } else {
                Mapa[Linha][Coluna] = '0';
                System.out.println("\nVoce acertou o mar, nenhum navio atingido. Restam " + (30 - Tentativas) + " tentativas.");
            }
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(x:"\nCoordenada inválida, por favor tente novamente\n");
        } catch (NumberFormatException e) {
            System.out.println(x:"\nFavor conceder coordenadas em valores numéricos válidos, tente novamente\n ");
        }
    } while (Verificador == 1);

    return acertos;
}

Run|Debug
public static void main(String[] args) {
    new Trabalho();
}
```

Visualização do Tabuleiro:

Início do jogo:

```
Iniciando jogo...

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~

Navios posicionados

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo Errou Navio:

```
Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
76

Voce acertou o mar, nenhum navio atingido. Restam 29 tentativas.

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ 0 ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo Acertou Navio:

```
Voce acertou o navio. Faltam: 9.

0 1 2 3 4 5 6 7 8
1 0 0 ~ ~ ~ ~ ~ 0
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ 0 ~
4 0 ~ ~ ~ ~ ~ ~ 0
5 ~ ~ 0 0 ~ 0 ~ ~
6 X ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ 0 ~ 0
8 ~ ~ ~ ~ ~ ~ 0 ~

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo Repetição de Coordenada:

```
Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
53

Coordenada indicada já foi atacada anteriormente...

Voce acertou o mar, nenhum navio atingido. Restam 18 tentativas.

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo PERDE:

```
VOCÊ PERDEU
APESAR DOS SEUS EXFORÇOS, A FROTA INIMIGA CONSEGUIU FUGIR

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ N ~ ~
2 ~ ~ ~ ~ ~ N N ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ N N N ~
5 ~ ~ ~ ~ ~ N ~ ~ ~
6 N ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ N ~ ~ ~ N

Deseja jogar novamente? (s) - Sim, (n) - Não
█
```

Exemplo, quando a coordenada fornecida contém 0 em alguma posição, o programa inválida as coordenadas fornecidas e pede para serem fornecidas novas coordenadas válidas:

```
Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
80

Coordenada inválida, por favor tente novamente

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo, quando as coordenadas fornecidas excedam 88, o programa os inválida e pede para serem fornecidas novas coordenadas:

```
Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
98

Coordenada inválida, por favor tente novamente

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
█
```

Exemplo, quando são fornecidas letras erroneamente como coordenadas, o programa identifica o erro e requisita novas coordenadas:

```
Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
gf
Favor conceder coordenadas em valores numéricos válidos, tente novamente

0 1 2 3 4 5 6 7 8
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~

Digite as coordenadas que deseja atacar (Favor informar a linha e a coluna, Exemplo: '12', Coluna = 1, Linha = 2):
```