

UNIVERSIDADE ANHEMBI MORUMBI

Campus Paulista - Noturno

Daniel de Sousa Ferreira Cardoso - RA 12524142630

Gabriel Rodrigues de Lima – RA 12524142939

Gian Carlo Tozzatti – RA 12524124240

Joaquim Guilherme Nunes Leal – RA 12524141595

Laryssa Rodrigues Vieira - RA 12524142179

PROJETO A3

Jogo Batalha Naval

GUERRA DOS MARES

**Trabalho de Avaliação 3 apresentado à
Universidade Anhembi Morumbi, na
matéria de Programação de Soluções
Computacionais.**

São Paulo

2024

Sumário

Introdução.....	3
Ideia do Projeto.....	4
Classe Navio	5
Classe Bomba	6
Classe Água.....	7
Classe Tabuleiro	8 e 9
Classe Pontos	10 e 11
Classe Jogo	12 a 15
Classe Objeto.....	16
Classe (Main)RodarJogo	17
Jogo na Prática	18 a 21
Conclusão	22
Referências.....	23

Introdução:

Neste trabalho , nos apresentaremos um jogo de batalha naval, desenvolvido totalmente em Java. A ideia surgiu a partir de uma conversa entre os integrantes do grupo. Pensamos em um jogo nostálgico e não tanto escolhido pelos outros grupos.

Ao pensarmos em batalha naval, automaticamente veio em nossa mente a batalha naval do programa Bom Dia e Companhia , do SBT.

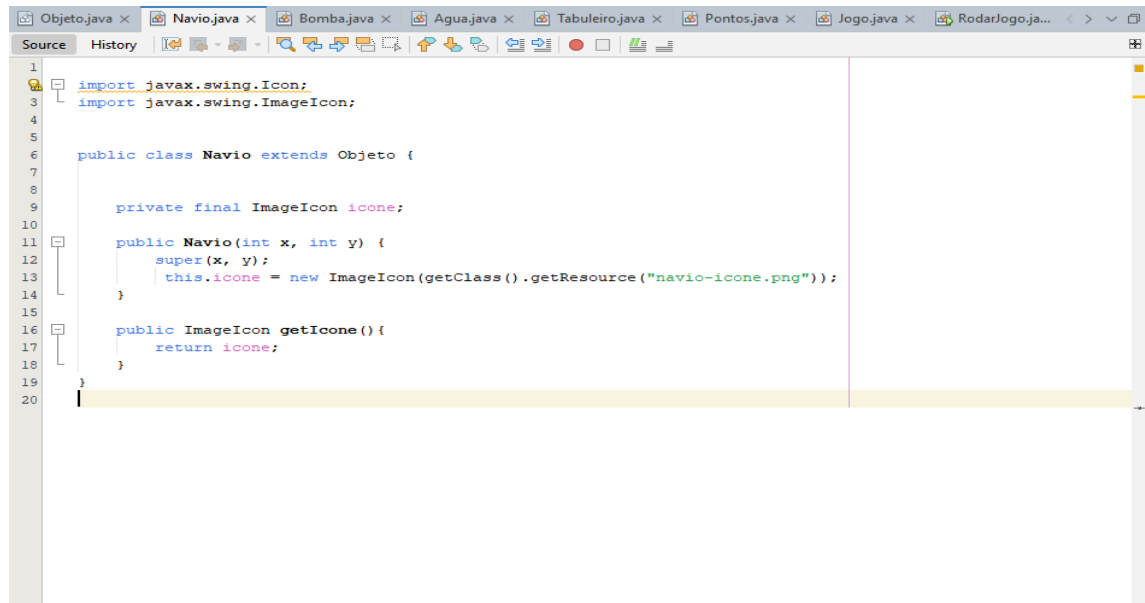
Entretanto, o jogo desenvolvido tem todas as características do nosso grupo, um jogo cheio de desafios, totalmente interativo , e divertido, fazendo com que os usuários fiquem completamente tidos durante a experiência do nosso game.

Dessa forma, nos apresentamos o Guerra dos Mares (Batalha Naval), que consiste num tabuleiro 5 x 5, onde seu objetivo é encontrar 3 navios espalhados pelo oceano, onde não se encontra apenas navios, mas, lugares perigosos, com sérios riscos de ser bombardeados.

Ideia do Projeto:

Apresentamos o Guerra dos Mares (Batalha Naval), um game que consiste num tabuleiro 5 x 5, onde seu objetivo é encontrar 3 navios espalhados pelo oceano, onde não se encontra apenas navios, mas, lugares perigosos, com sérios riscos de ser bombardeados.

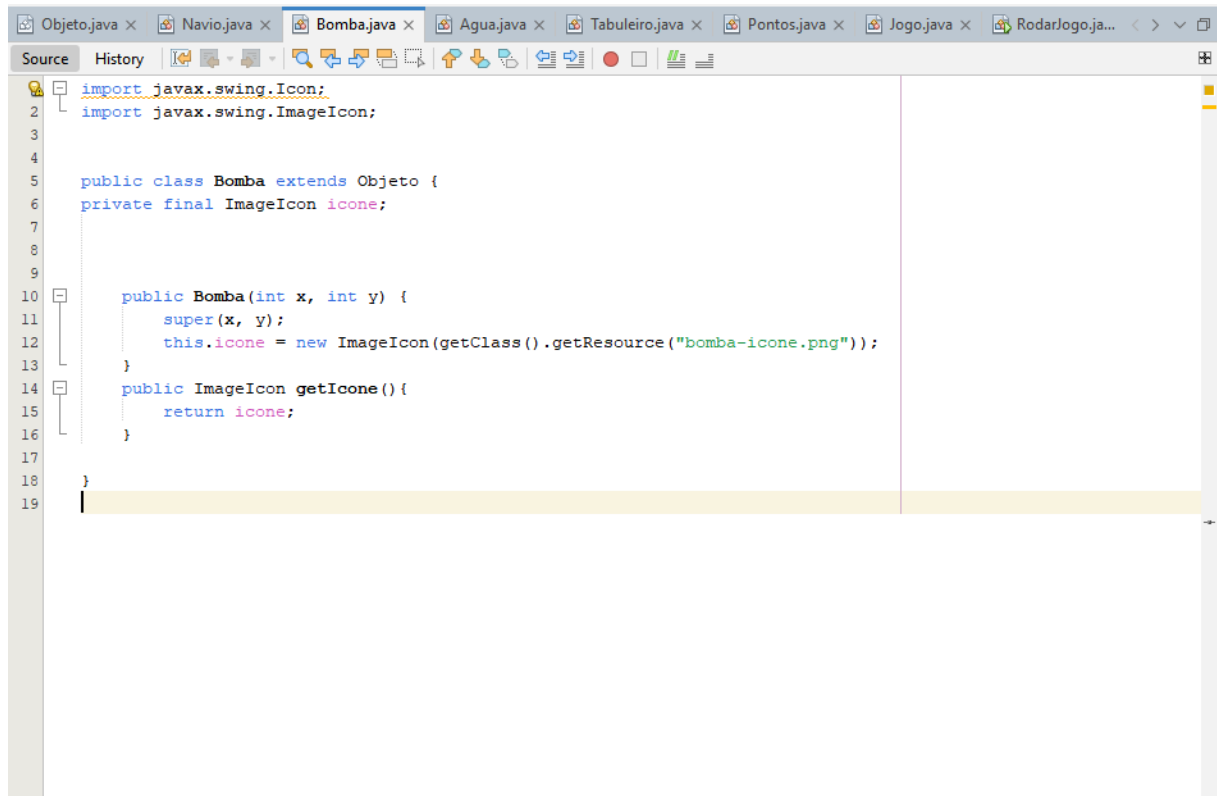
Classe Navio:



```
1  import javax.swing.Icon;
2  import javax.swing.ImageIcon;
3
4
5
6  public class Navio extends Objeto {
7
8
9      private final ImageIcon icon;
10
11      public Navio(int x, int y) {
12          super(x, y);
13          this.icon = new ImageIcon(getClass().getResource("navio-icone.png"));
14      }
15
16      public ImageIcon getIcon() {
17          return icon;
18      }
19  }
20
```

Este código está definindo a classe nomeada **Navio**, ela estende a classe Objeto. A classe Navio tem um construtor onde aceita coordenadas x e y como parâmetros e inicializa o ícone do navio usando a classe ImageIcon com o recurso "navio-icone.png". O método getIcon(), retorna o ícone do navio.

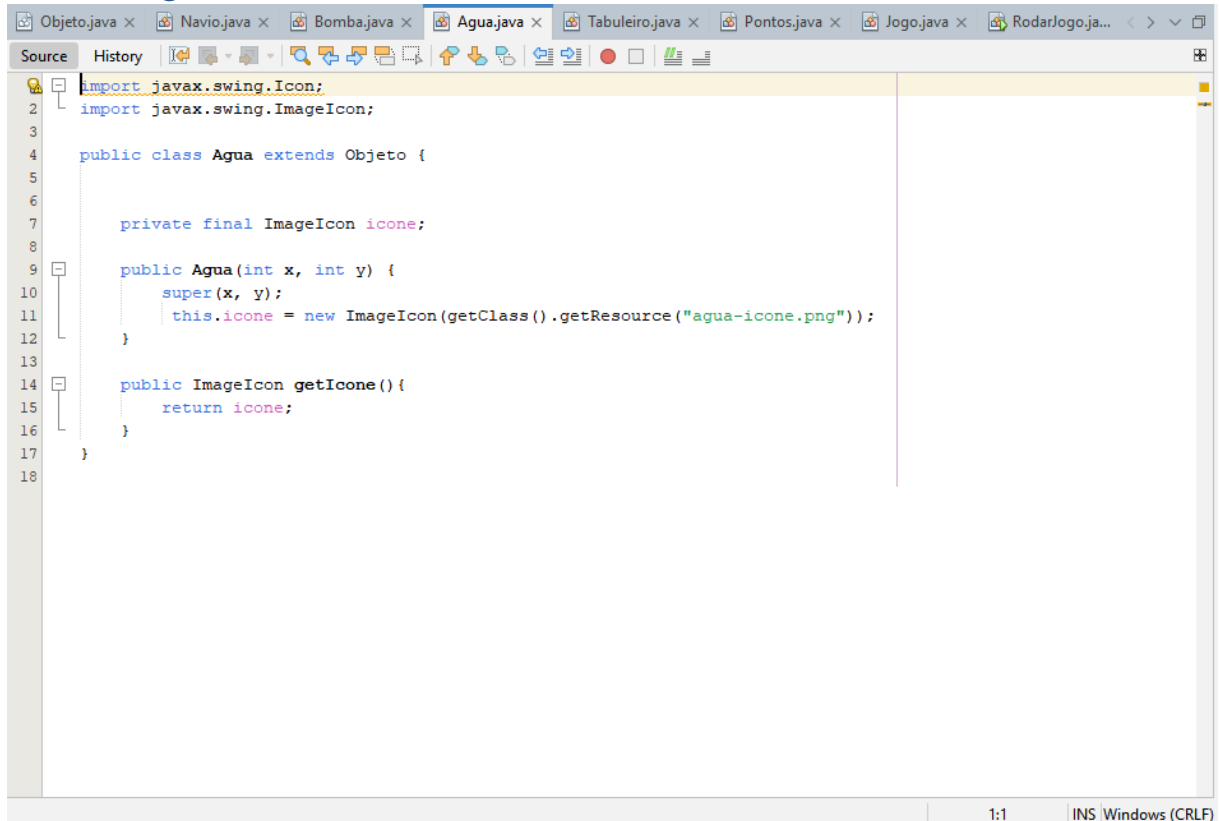
Classe Bomba:



```
1  import javax.swing.Icon;
2  import javax.swing.ImageIcon;
3
4
5  public class Bomba extends Objeto {
6      private final ImageIcon icone;
7
8
9
10     public Bomba(int x, int y) {
11         super(x, y);
12         this.icone = new ImageIcon(getClass().getResource("bomba-icone.png"));
13     }
14     public ImageIcon getIcon() {
15         return icone;
16     }
17
18 }
19
```

Este código define uma classe chamada **Bomba** que estende a classe Objeto. O construtor da classe Bomba aceita coordenadas x e y como parâmetros e inicializa o ícone da bomba usando a classe ImageIcon com o recurso "bomba-icone.png". O método getIcon() retorna o ícone da bomba.

Classe Água:

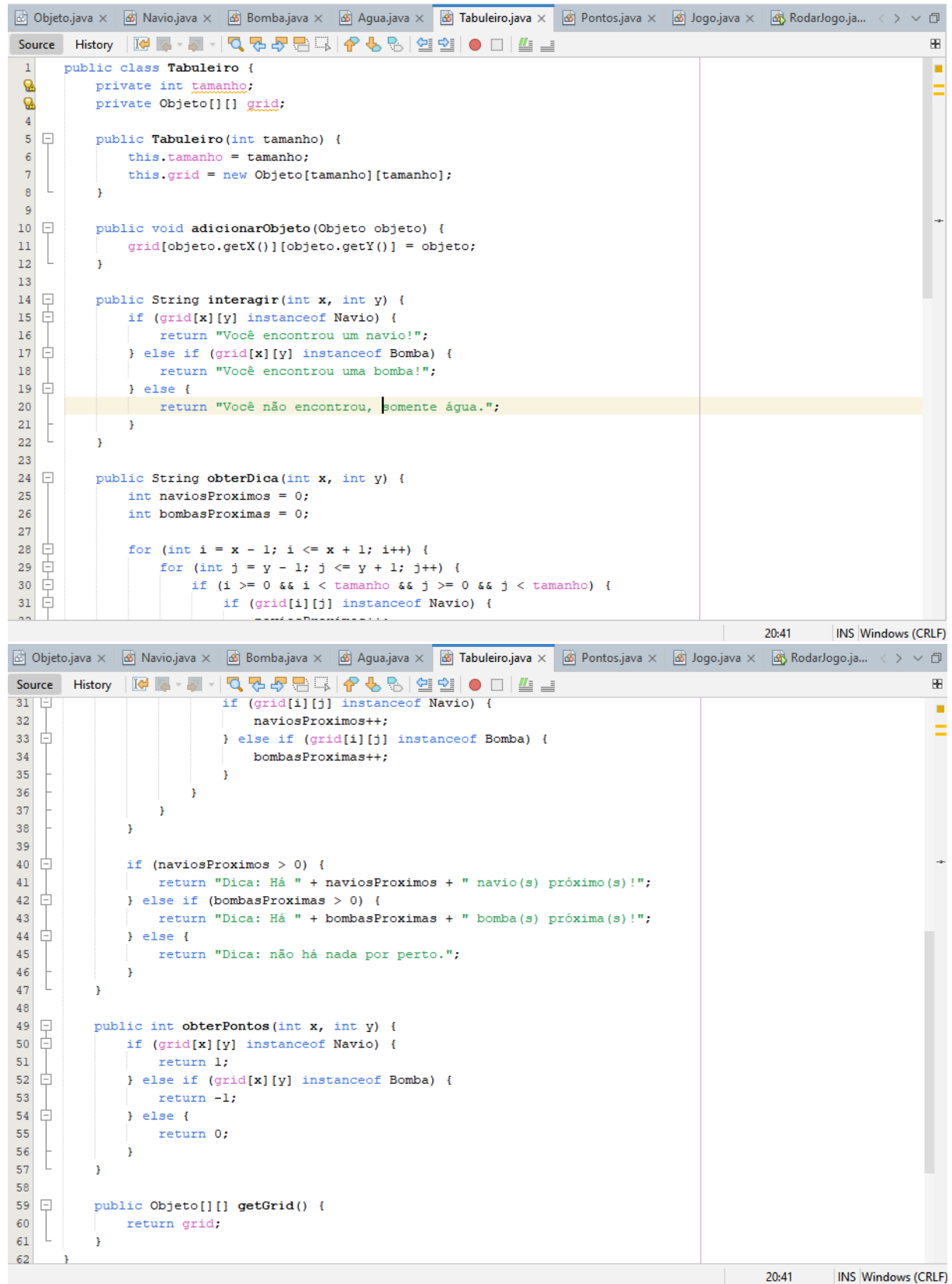
The image shows a screenshot of a Java IDE with multiple tabs open. The active tab is 'Água.java'. The code in the editor is as follows:

```
1 import javax.swing.Icon;
2 import javax.swing.ImageIcon;
3
4 public class Agua extends Objeto {
5
6     private final ImageIcon icone;
7
8     public Agua(int x, int y) {
9         super(x, y);
10        this.icone = new ImageIcon(getClass().getResource("agua-icone.png"));
11    }
12
13    public ImageIcon getIcon() {
14        return icone;
15    }
16 }
17
18
```

The IDE interface includes a toolbar with various icons for editing and running code, and a status bar at the bottom showing '1:1' and 'INS Windows (CRLF)'.

Este código define uma classe chamada **Água** que estende a classe **Objeto**. O construtor da classe **Água** aceita coordenadas **x** e **y** como parâmetros e inicializa o ícone da água usando a classe **ImageIcon** com o recurso "agua-icone.png". O método **getIcon()** retorna o ícone da água.

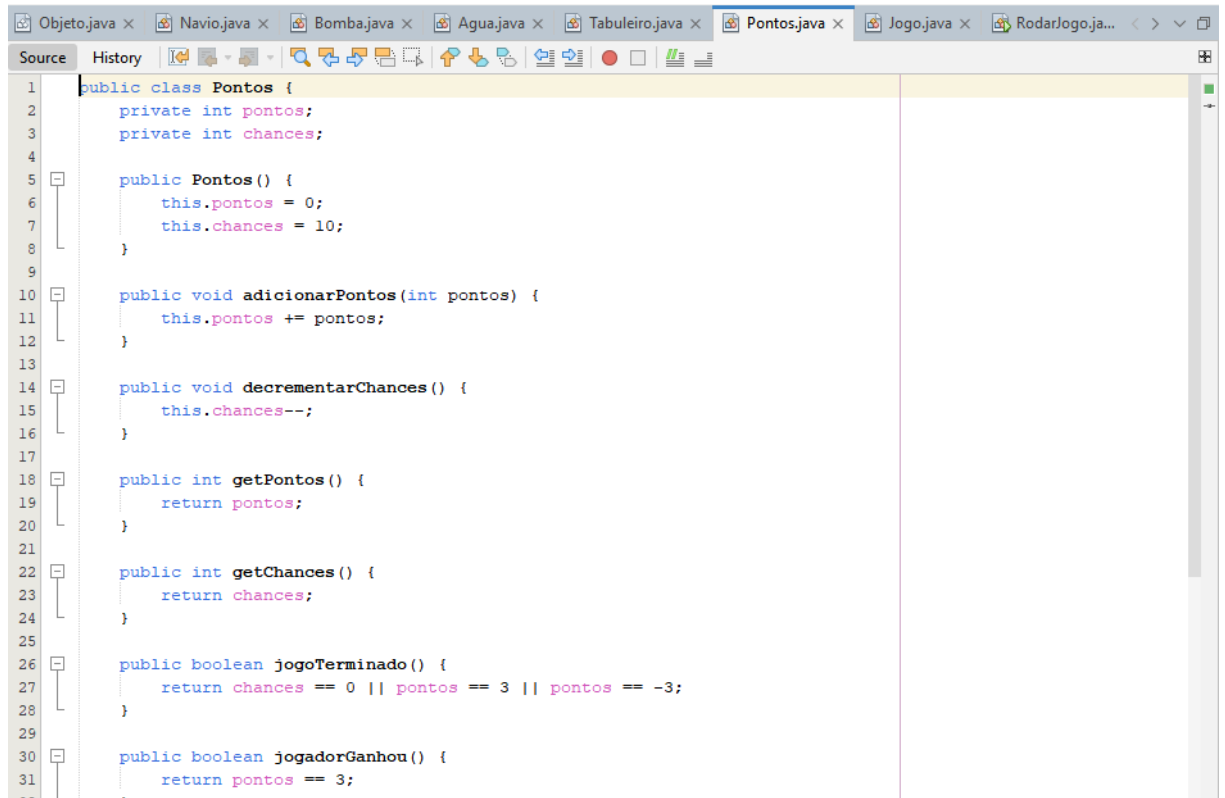
Classe Tabuleiro:



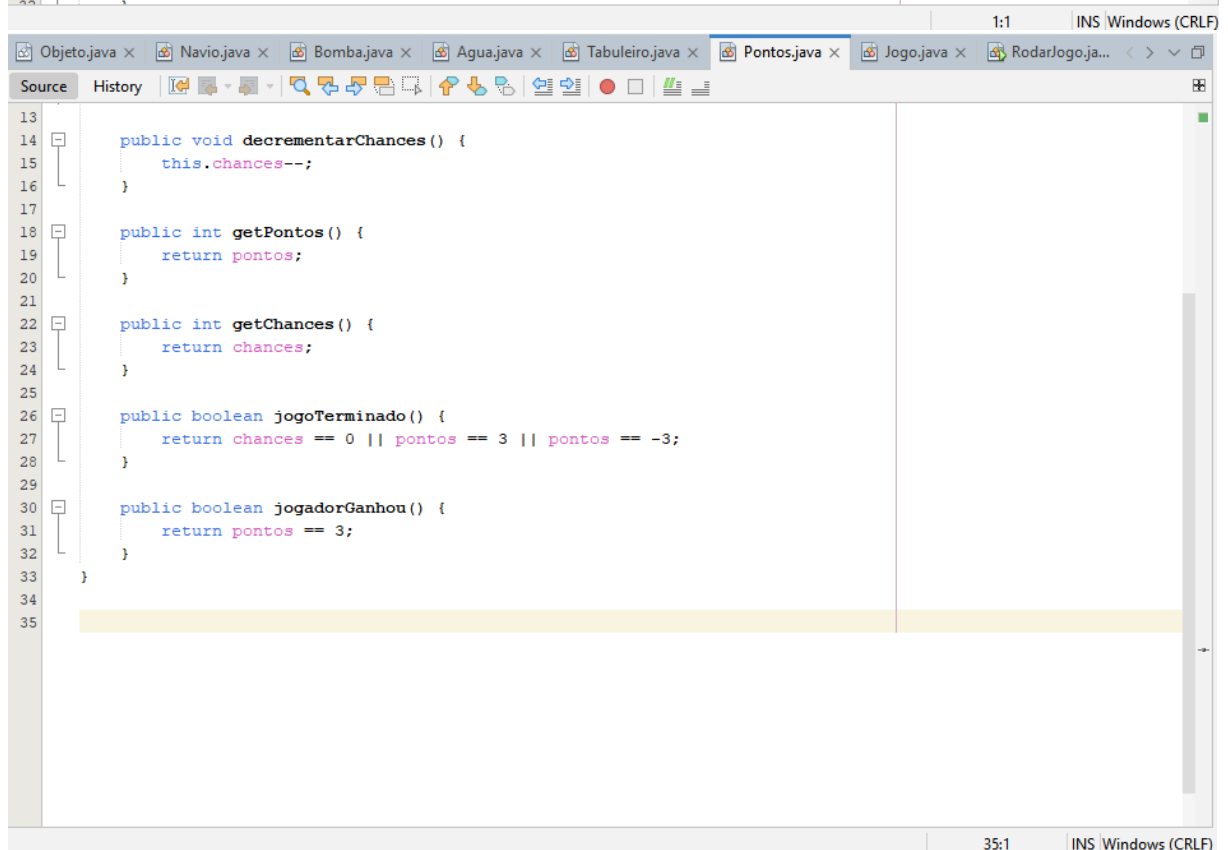
```
1 public class Tabuleiro {
2     private int tamanho;
3     private Objeto[][] grid;
4
5     public Tabuleiro(int tamanho) {
6         this.tamanho = tamanho;
7         this.grid = new Objeto[tamanho][tamanho];
8     }
9
10    public void adicionarObjeto(Objeto objeto) {
11        grid[objeto.getX()][objeto.getY()] = objeto;
12    }
13
14    public String interagir(int x, int y) {
15        if (grid[x][y] instanceof Navio) {
16            return "Você encontrou um navio!";
17        } else if (grid[x][y] instanceof Bomba) {
18            return "Você encontrou uma bomba!";
19        } else {
20            return "Você não encontrou, somente água.";
21        }
22    }
23
24    public String obterDica(int x, int y) {
25        int naviosProximos = 0;
26        int bombasProximas = 0;
27
28        for (int i = x - 1; i <= x + 1; i++) {
29            for (int j = y - 1; j <= y + 1; j++) {
30                if (i >= 0 && i < tamanho && j >= 0 && j < tamanho) {
31                    if (grid[i][j] instanceof Navio) {
32                        naviosProximos++;
33                    } else if (grid[i][j] instanceof Bomba) {
34                        bombasProximas++;
35                    }
36                }
37            }
38        }
39
40        if (naviosProximos > 0) {
41            return "Dica: Há " + naviosProximos + " navio(s) próximo(s)!";
42        } else if (bombasProximas > 0) {
43            return "Dica: Há " + bombasProximas + " bomba(s) próxima(s)!";
44        } else {
45            return "Dica: não há nada por perto.";
46        }
47    }
48
49    public int obterPontos(int x, int y) {
50        if (grid[x][y] instanceof Navio) {
51            return 1;
52        } else if (grid[x][y] instanceof Bomba) {
53            return -1;
54        } else {
55            return 0;
56        }
57    }
58
59    public Objeto[][] getGrid() {
60        return grid;
61    }
62 }
```


Esta classe **Tabuleiro** representa um tabuleiro de jogo onde os objetos (Navio, Bomba e Água) podem ser colocados. Ele fornece métodos para interagir com o tabuleiro, como adicionar objetos, interagir em uma posição específica, obter dicas sobre objetos próximos e obter pontos associados a uma determinada posição no tabuleiro.

Classe Pontos:



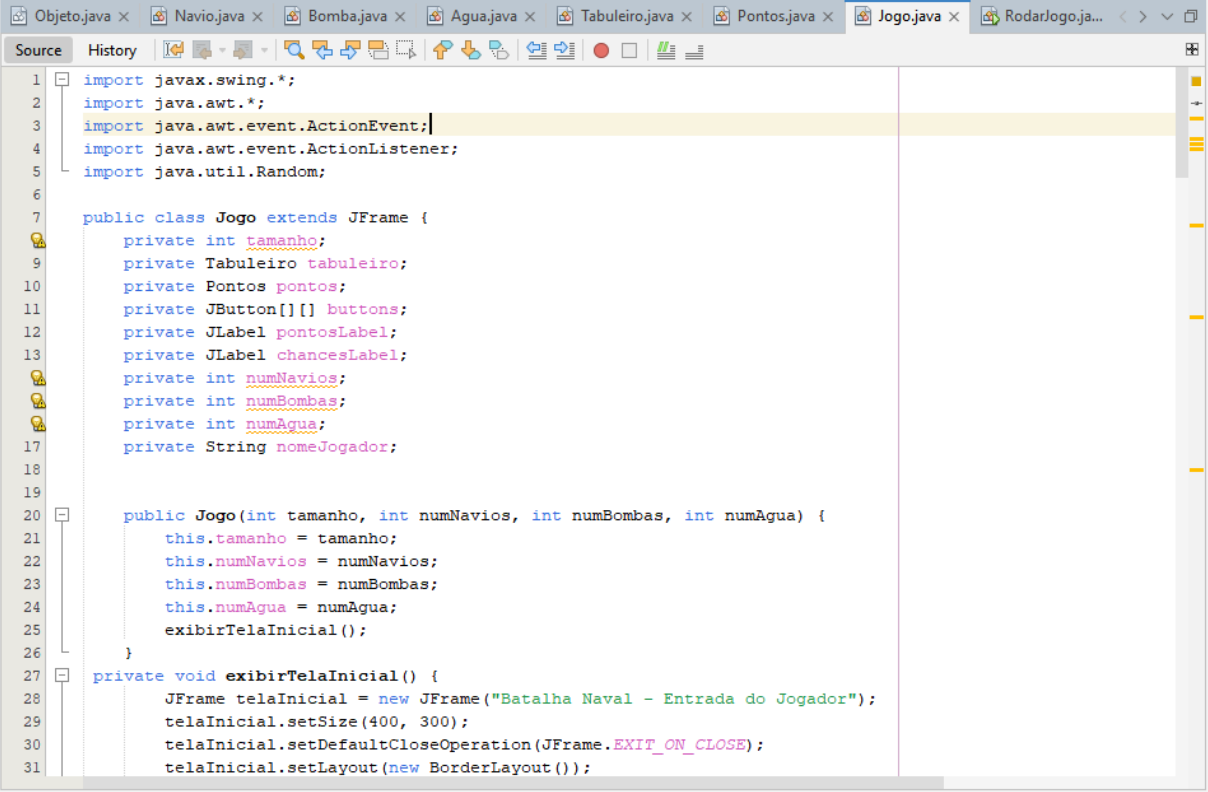
```
1 public class Pontos {
2     private int pontos;
3     private int chances;
4
5     public Pontos() {
6         this.pontos = 0;
7         this.chances = 10;
8     }
9
10    public void adicionarPontos(int pontos) {
11        this.pontos += pontos;
12    }
13
14    public void decrementarChances() {
15        this.chances--;
16    }
17
18    public int getPontos() {
19        return pontos;
20    }
21
22    public int getChances() {
23        return chances;
24    }
25
26    public boolean jogoTerminado() {
27        return chances == 0 || pontos == 3 || pontos == -3;
28    }
29
30    public boolean jogadorGanhou() {
31        return pontos == 3;
32    }
33 }
```



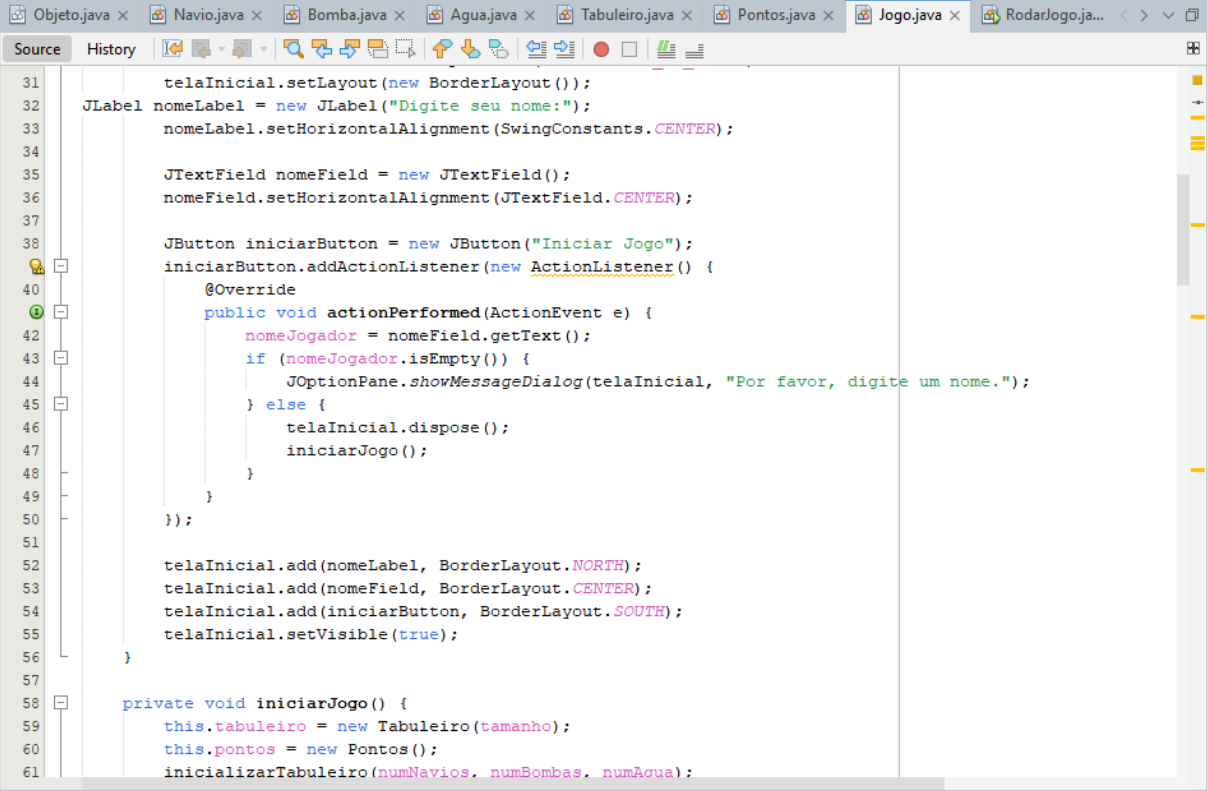
```
13
14    public void decrementarChances() {
15        this.chances--;
16    }
17
18    public int getPontos() {
19        return pontos;
20    }
21
22    public int getChances() {
23        return chances;
24    }
25
26    public boolean jogoTerminado() {
27        return chances == 0 || pontos == 3 || pontos == -3;
28    }
29
30    public boolean jogadorGanhou() {
31        return pontos == 3;
32    }
33 }
34
35
```

Essa classe **Pontos** controla os pontos e as chances em um jogo. Ela mantém o controle dos pontos do jogador e do número de chances restantes. Os métodos permitem adicionar pontos, decrementar as chances, obter o número de pontos e de chances, verificar se o jogo terminou (se o jogador ficou sem chances ou atingiu 3 pontos ou -3 pontos) e se o jogador ganhou (atingiu 3 pontos).

Classe Jogo:



```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.Random;
6
7 public class Jogo extends JFrame {
8     private int tamanho;
9     private Tabuleiro tabuleiro;
10    private Pontos pontos;
11    private JButton[][] buttons;
12    private JLabel pontosLabel;
13    private JLabel chancesLabel;
14    private int numNavios;
15    private int numBombas;
16    private int numAgua;
17    private String nomeJogador;
18
19    public Jogo(int tamanho, int numNavios, int numBombas, int numAgua) {
20        this.tamanho = tamanho;
21        this.numNavios = numNavios;
22        this.numBombas = numBombas;
23        this.numAgua = numAgua;
24        exibirTelaInicial();
25    }
26
27    private void exibirTelaInicial() {
28        JFrame telaInicial = new JFrame("Batalha Naval - Entrada do Jogador");
29        telaInicial.setSize(400, 300);
30        telaInicial.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31        telaInicial.setLayout(new BorderLayout());
32
33        JLabel nomeLabel = new JLabel("Digite seu nome:");
34        nomeLabel.setHorizontalAlignment(SwingConstants.CENTER);
35
36        JTextField nomeField = new JTextField();
37        nomeField.setHorizontalAlignment(JTextField.CENTER);
38
39        JButton iniciarButton = new JButton("Iniciar Jogo");
40        iniciarButton.addActionListener(new ActionListener() {
41            @Override
42            public void actionPerformed(ActionEvent e) {
43                nomeJogador = nomeField.getText();
44                if (nomeJogador.isEmpty()) {
45                    JOptionPane.showMessageDialog(telaInicial, "Por favor, digite um nome.");
46                } else {
47                    telaInicial.dispose();
48                    iniciarJogo();
49                }
50            }
51        });
52
53        telaInicial.add(nomeLabel, BorderLayout.NORTH);
54        telaInicial.add(nomeField, BorderLayout.CENTER);
55        telaInicial.add(iniciarButton, BorderLayout.SOUTH);
56        telaInicial.setVisible(true);
57    }
58
59    private void iniciarJogo() {
60        this.tabuleiro = new Tabuleiro(tamanho);
61        this.pontos = new Pontos();
62        inicializarTabuleiro(numNavios, numBombas, numAgua);
63    }
```



```
32 telaInicial.setLayout(new BorderLayout());
33 JLabel nomeLabel = new JLabel("Digite seu nome:");
34 nomeLabel.setHorizontalAlignment(SwingConstants.CENTER);
35
36 JTextField nomeField = new JTextField();
37 nomeField.setHorizontalAlignment(JTextField.CENTER);
38
39 JButton iniciarButton = new JButton("Iniciar Jogo");
40 iniciarButton.addActionListener(new ActionListener() {
41     @Override
42     public void actionPerformed(ActionEvent e) {
43         nomeJogador = nomeField.getText();
44         if (nomeJogador.isEmpty()) {
45             JOptionPane.showMessageDialog(telaInicial, "Por favor, digite um nome.");
46         } else {
47             telaInicial.dispose();
48             iniciarJogo();
49         }
50     }
51 });
52
53 telaInicial.add(nomeLabel, BorderLayout.NORTH);
54 telaInicial.add(nomeField, BorderLayout.CENTER);
55 telaInicial.add(iniciarButton, BorderLayout.SOUTH);
56 telaInicial.setVisible(true);
57
58 private void iniciarJogo() {
59     this.tabuleiro = new Tabuleiro(tamanho);
60     this.pontos = new Pontos();
61     inicializarTabuleiro(numNavios, numBombas, numAgua);
62 }
```

```

Objeto.java x Navio.java x Bomba.java x Agua.java x Tabuleiro.java x Pontos.java x Jogo.java x RodarJogo.ja...
Source History
61      inicializarTabuleiro(numNavios, numBombas, numAgua);
62      initComponents();
63  }
64
65
66  private void inicializarTabuleiro(int numNavios, int numBombas, int numAgua) {
67      Random rand = new Random();
68      int colocados = 0;
69      while (colocados < numNavios) {
70          int x = rand.nextInt(tamanho);
71          int y = rand.nextInt(tamanho);
72          if (tabuleiro.getGrid()[x][y] == null) {
73              tabuleiro.adicionarObjeto(new Navio(x, y));
74              colocados++;
75          }
76      }
77
78      colocados = 0;
79      while (colocados < numBombas) {
80          int x = rand.nextInt(tamanho);
81          int y = rand.nextInt(tamanho);
82          if (tabuleiro.getGrid()[x][y] == null) {
83              tabuleiro.adicionarObjeto(new Bomba(x, y));
84              colocados++;
85          }
86      }
87      colocados = 0;
88      while (colocados < numAgua) {
89          int x = rand.nextInt(tamanho);
90          int y = rand.nextInt(tamanho);
91          if (tabuleiro.getGrid()[x][y] == null) {

```

```

3:35 | INS Windows (CRLF)
Objeto.java x Navio.java x Bomba.java x Agua.java x Tabuleiro.java x Pontos.java x Jogo.java x RodarJogo.ja...
Source History
91      if (tabuleiro.getGrid()[x][y] == null) {
92          tabuleiro.adicionarObjeto(new Agua(x, y));
93          colocados++;
94      }
95  }
96  }
97
98  private void initComponents() {
99      setTitle("Batalha Naval");
100      setSize(600, 600);
101      setLayout(new BorderLayout());
102
103      JPanel gridPanel = new JPanel(new GridLayout(tamanho, tamanho));
104      JButton[] buttons = new JButton[tamanho][tamanho];
105      for (int i = 0; i < tamanho; i++) {
106          for (int j = 0; j < tamanho; j++) {
107              final int x = i;
108              final int y = j;
109              buttons[i][j] = new JButton();
110              buttons[i][j].addActionListener(new ActionListener() {
111                  @Override
112                  public void actionPerformed(ActionEvent e) {
113                      jogar(x, y);
114                  }
115              });
116              gridPanel.add(buttons[i][j]);
117          }
118      }
119
120      JPanel statusPanel = new JPanel(new GridLayout(1, 2));
121      pontosLabel = new JLabel("Pontos: " + pontos.getPontos());

```

```

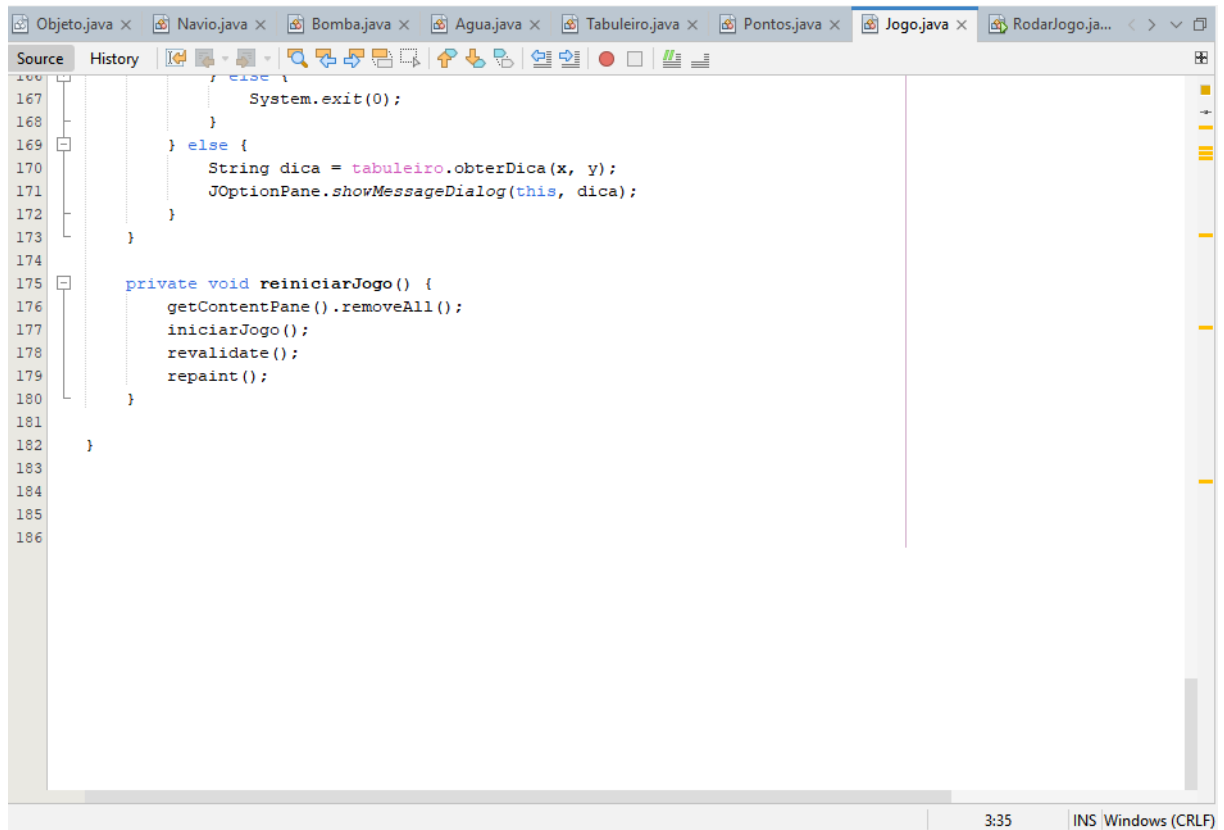
Objeto.java x Navio.java x Bomba.java x Agua.java x Tabuleiro.java x Pontos.java x Jogo.java x RodarJogo.ja...
Source History
119
120     JPanel statusPanel = new JPanel(new GridLayout(1, 2));
121     pontosLabel = new JLabel("Pontos: " + pontos.getPontos());
122     pontosLabel.setHorizontalAlignment(SwingConstants.CENTER);
123     chancesLabel = new JLabel("Chances: " + pontos.getChances());
124     chancesLabel.setHorizontalAlignment(SwingConstants.CENTER);
125     statusPanel.add(pontosLabel);
126     statusPanel.add(chancesLabel);
127
128     add(gridPanel, BorderLayout.CENTER);
129     add(statusPanel, BorderLayout.SOUTH);
130
131     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
132     setVisible(true);
133 }
134
135 private void jogar(int x, int y) {
136     Objeto objeto = tabuleiro.getGrid()[x][y];
137     String resultado = tabuleiro.interagir(x, y);
138     int pontosObtidos = tabuleiro.obterPontos(x, y);
139     pontos.adicionarPontos(pontosObtidos);
140     pontos.decrementarChances();
141     if (pontosObtidos > 0) {
142         buttons[x][y].setText("Navio");
143         buttons[x][y].setIcon(((Navio)objeto).getIcone());
144     } else if (pontosObtidos < 0) {
145         buttons[x][y].setText("Bomba");
146         buttons[x][y].setIcon(((Bomba)objeto).getIcone());
147     } else {
148         buttons[x][y].setText("Água");
149         buttons[x][y].setIcon(((Agua)objeto).getIcone());

```

```

3:35 | INS Windows (CRLF)
Objeto.java x Navio.java x Bomba.java x Agua.java x Tabuleiro.java x Pontos.java x Jogo.java x RodarJogo.ja...
Source History
149         buttons[x][y].setIcon(((Agua)objeto).getIcone());
150
151     }
152     buttons[x][y].setEnabled(false);
153     pontosLabel.setText("Pontos: " + pontos.getPontos());
154     chancesLabel.setText("Chances: " + pontos.getChances());
155     JOptionPane.showMessageDialog(this, resultado);
156
157     if (pontos.jogoTerminado()) {
158         if (pontos.jogadorGanhou()) {
159             JOptionPane.showMessageDialog(this, "Parabéns! " + nomeJogador + " Você venceu o jogo!");
160         } else {
161             JOptionPane.showMessageDialog(this, "Fim de jogo!!! " + nomeJogador + " Você perdeu.");
162         }
163         int resposta = JOptionPane.showConfirmDialog(this, "Deseja jogar novamente?", "Jogo Terminado",
164             if (resposta == JOptionPane.YES_OPTION) {
165                 reiniciarJogo();
166             } else {
167                 System.exit(0);
168             }
169     } else {
170         String dica = tabuleiro.obterDica(x, y);
171         JOptionPane.showMessageDialog(this, dica);
172     }
173 }
174
175 private void reiniciarJogo() {
176     getContentPane().removeAll();
177     iniciarJogo();
178     revalidate();
179     repaint();

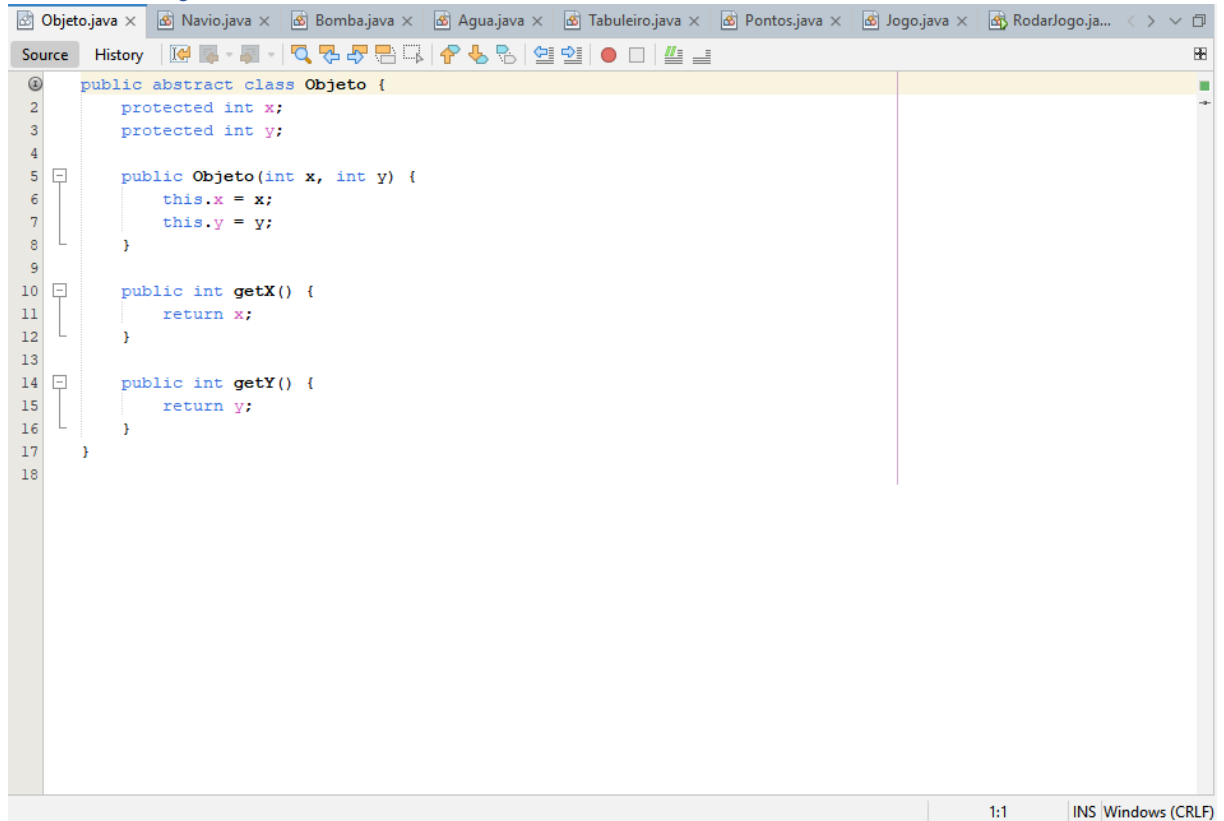
```



```
166         } else {
167             System.exit(0);
168         }
169     } else {
170         String dica = tabuleiro.obterDica(x, y);
171         JOptionPane.showMessageDialog(this, dica);
172     }
173 }
174
175 private void reiniciarJogo() {
176     getContentPane().removeAll();
177     iniciarJogo();
178     revalidate();
179     repaint();
180 }
181
182 }
183
184
185
186
```

Esse é a classe **Jogo** que estende o JFrame onde gera a tela inicial e a interface do jogo. Vários atributos são declarados para manter o estado do jogo, como o tamanho do tabuleiro, número de navios, bombas, células de água, o tabuleiro em si, pontos

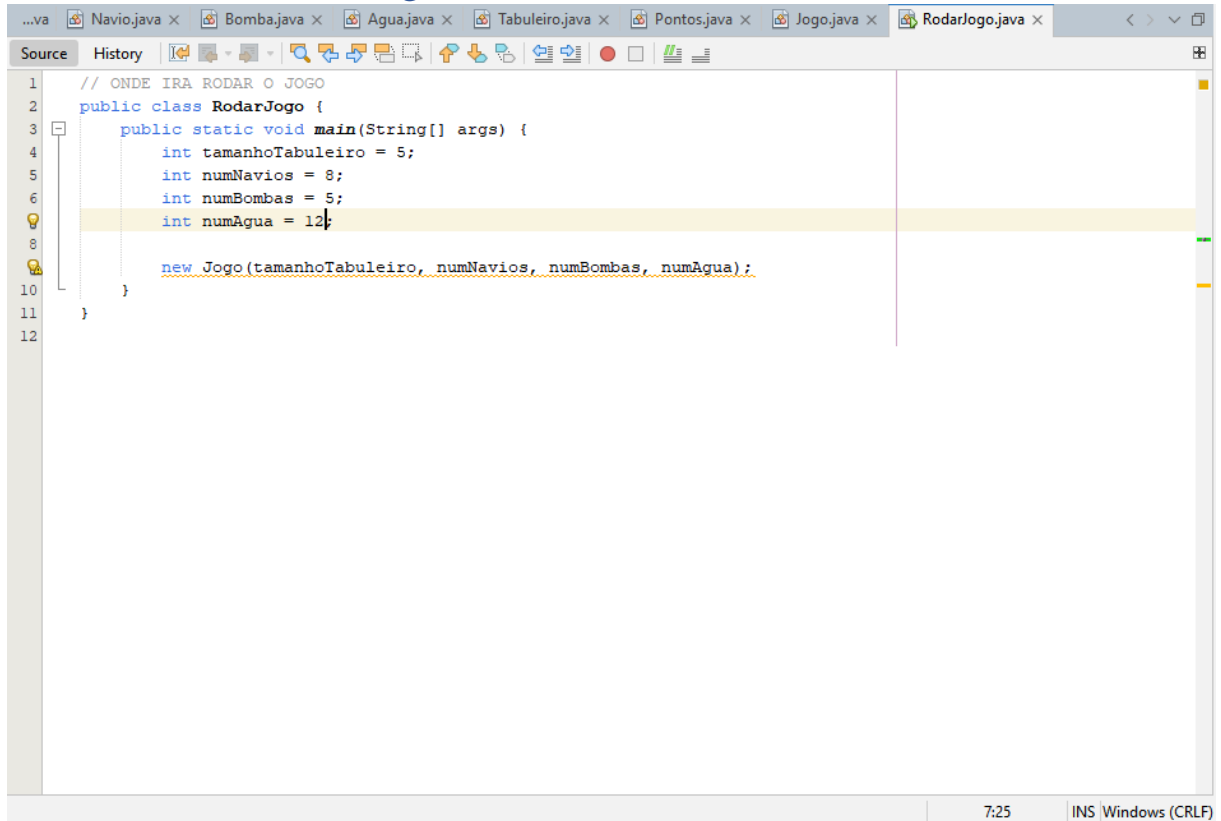
Classe Objeto:



```
1 public abstract class Objeto {
2     protected int x;
3     protected int y;
4
5     public Objeto(int x, int y) {
6         this.x = x;
7         this.y = y;
8     }
9
10    public int getX() {
11        return x;
12    }
13
14    public int getY() {
15        return y;
16    }
17 }
18
```

Essa classe é uma classe abstrata, o que significa que ela não pode ser instanciada diretamente. Serve como uma base para outras classes que representam objetos específicos no jogo, como Navio, Bomba e Água. Ela contém métodos para obter as coordenadas x e y do objeto.

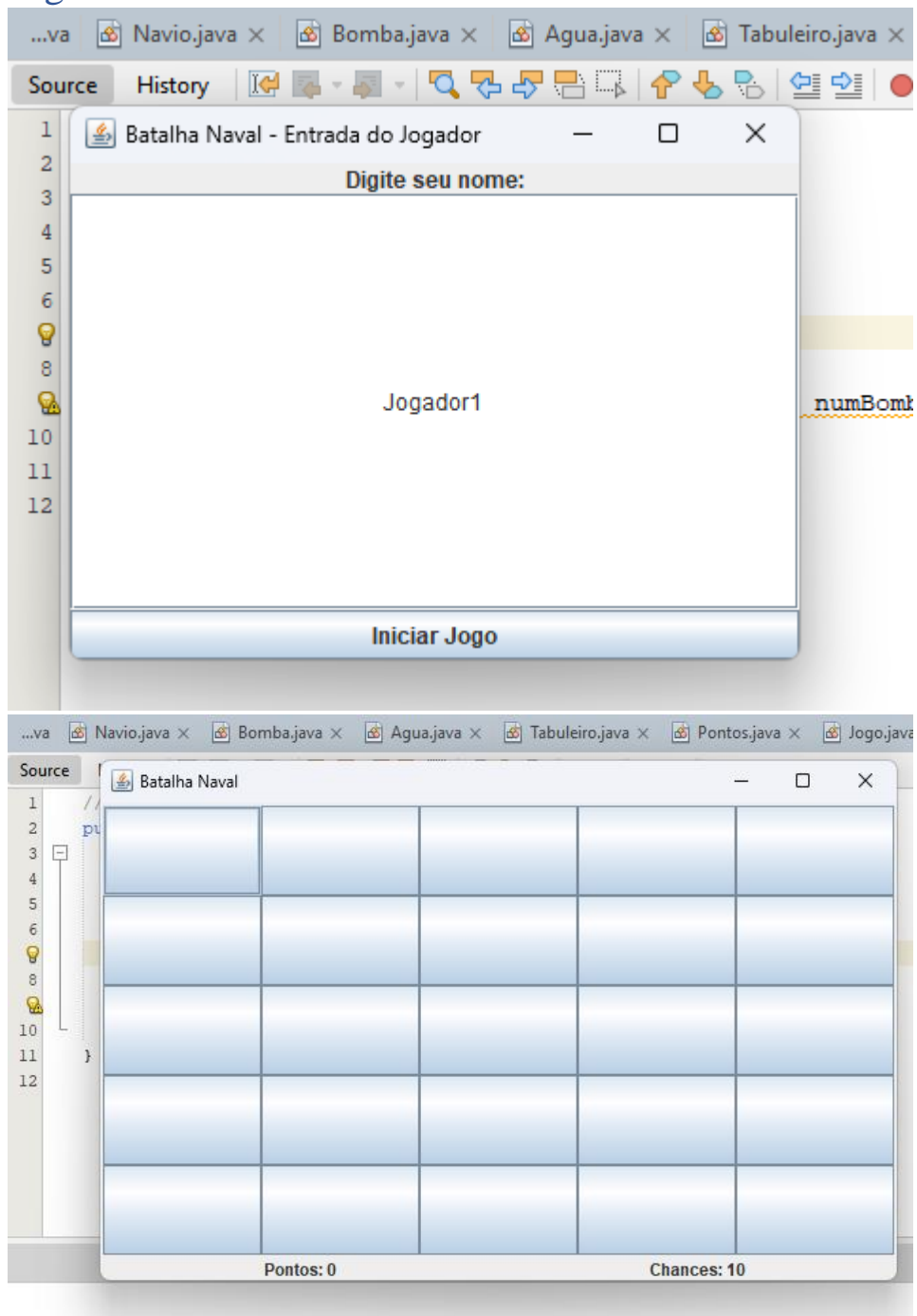
Classe (Main) RodarJogo:

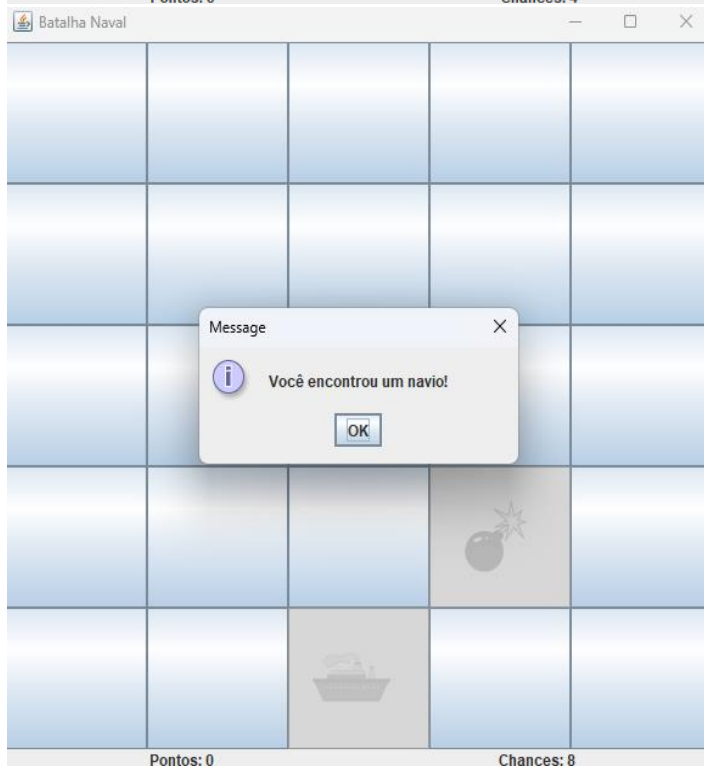
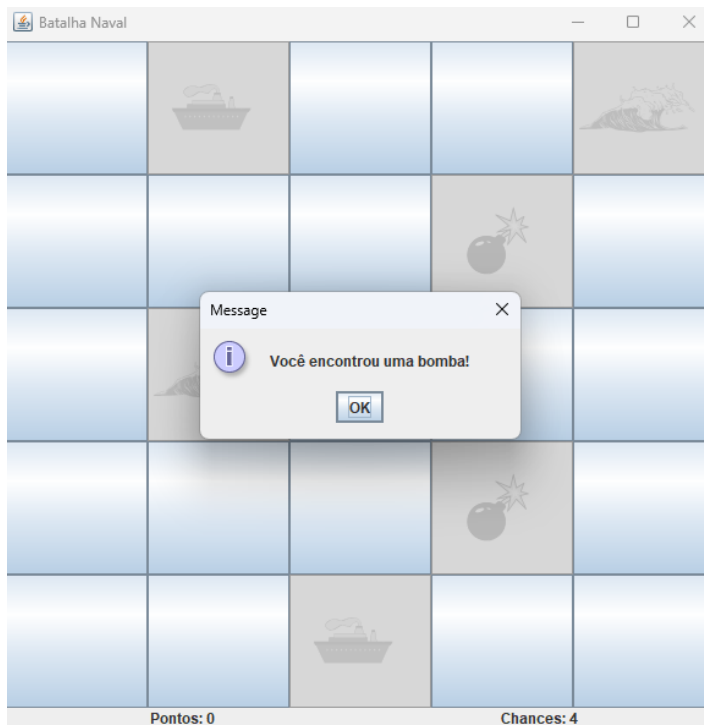


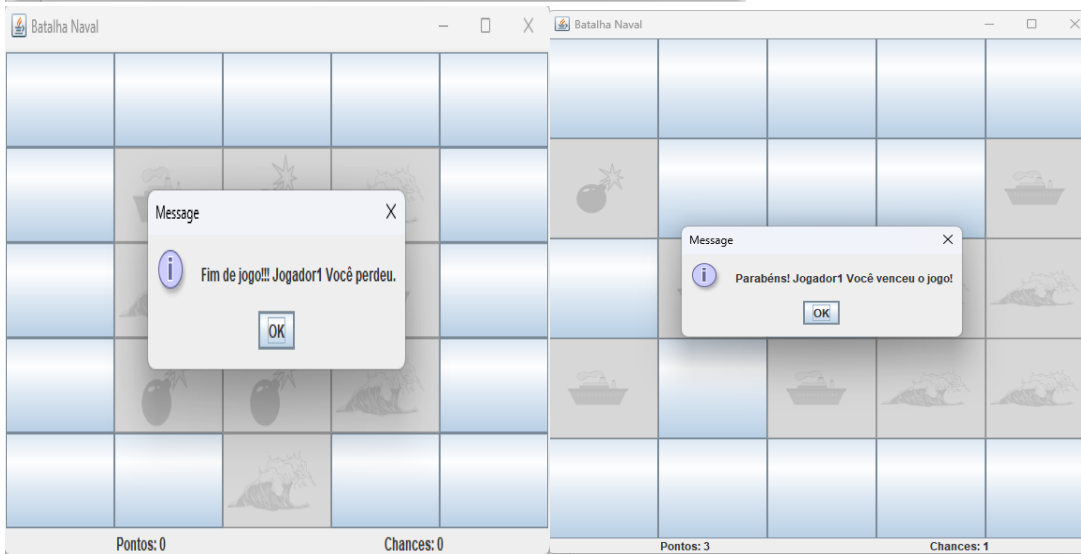
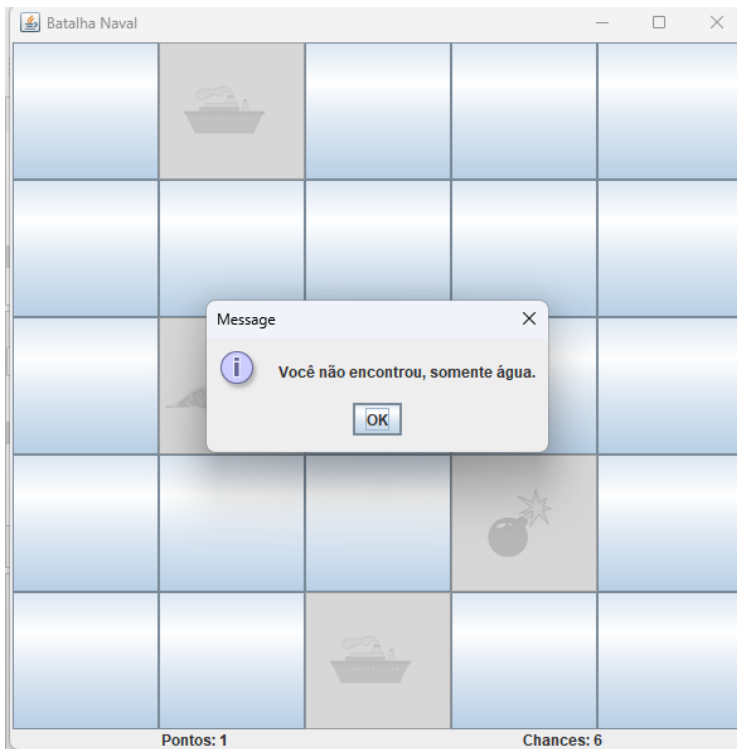
```
1 // ONDE IRA RODAR O JOGO
2 public class RodarJogo {
3     public static void main(String[] args) {
4         int tamanhoTabuleiro = 5;
5         int numNavios = 8;
6         int numBombas = 5;
7         int numAgua = 12;
8
9         new Jogo(tamanhoTabuleiro, numNavios, numBombas, numAgua);
10    }
11 }
12
```

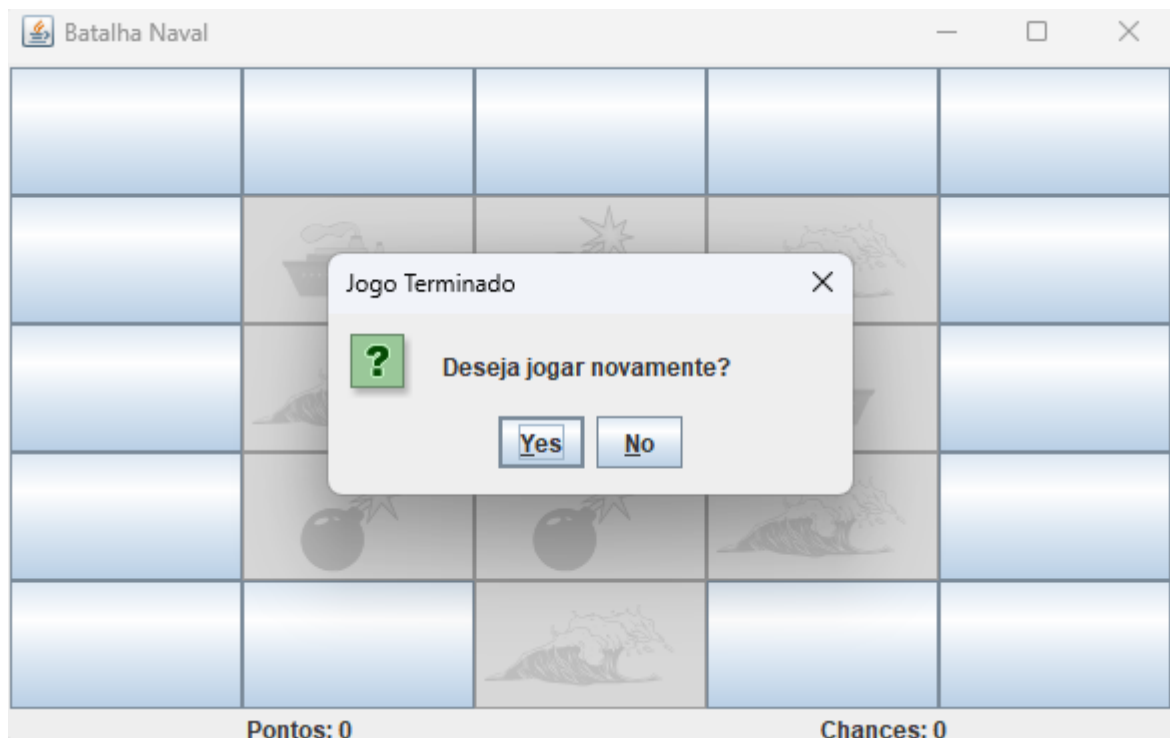
Essa classe **RodarJogo** possui o método main que cria uma instância da classe Jogo com os parâmetros definidos. Quando o programa é executado, isso iniciará o jogo com as configurações especificadas.

Jogo na Prática:









O jogo inicia com uma tela de entrada, um local que serve para o jogador digitar o nome desejado, após isso você iniciara o jogo, onde é gerado um tabuleiro 5x5, onde se encontram 8 navios, 5 bombas e 12 águas que serão mudadas de forma aleatória a cada rodada, seu objetivo é fazer 3 pontos ou (achar 3 navios), se achar um navio ganha um ponto, se você encontrar bomba, perde um ponto, caso encontre água, a rodada passará em branco sem nenhum ponto. Em caso de vitória o jogo vai printar uma imagem com a mensagem que “Parabéns! Jogador Você Venceu o Jogo”, já em caso de derrota irá printar a imagem com mensagem “Fim de Jogo! Jogador Você Perdeu.”. Após isso aparecera uma imagem com mensagem “Deseja jogar novamente?”, onde você terá a opção de jogar novamente. Esse é o **Guerra dos Mares**.

Conclusão:

Em suma, o desenvolvimento deste jogo de Batalha Naval em Java com interface gráfica foi uma experiência desafiadora e gratificante. Ao longo do processo, pudemos aplicar conceitos fundamentais de programação orientada a objetos, polimorfismo, herança e interface gráfica para criar uma experiência interativa e envolvente para os jogadores.

Além disso, entendemos que a criação desse jogo, foi de extrema importância para o desenvolvimento e aprimoramento de habilidades de resolução de problemas e colaboração em equipe.

Esperamos que os usuários desfrutem desse projeto feito com carinho e dedicação dos criadores.

Referências:

- <https://www.javaprogressivo.net/2012/09/jogo-batalha-naval-em-java.html?m=1>
- <https://pt.stackoverflow.com/questions/152229/como-otimizar-o-jogo-batalha-naval-em-java>
- https://youtu.be/qTPBqgRJT_g?si=OAXRsBtXtqeJPEZz
- <https://youtu.be/PUiYV-36LIC?si=kIDQLWcUhemcz9PE>
- <https://youtu.be/tnL3GPv5VsQ?si=PaU0AGILecPbmeq1>
- https://youtu.be/iSI6l1eoCyo?si=D4_0zb4q3OT3UuoK
- <https://youtu.be/PearQPnNzEE?si=o1Z3TPH1cK9laHpu>
- <https://youtu.be/FtVtCemBe9s?si=7JqK9DEMxwZY4xPv>