Trabalho de Programação Orientada a Objetos

07/10/2019

Prof. Ivan Mathias Filho

Introdução

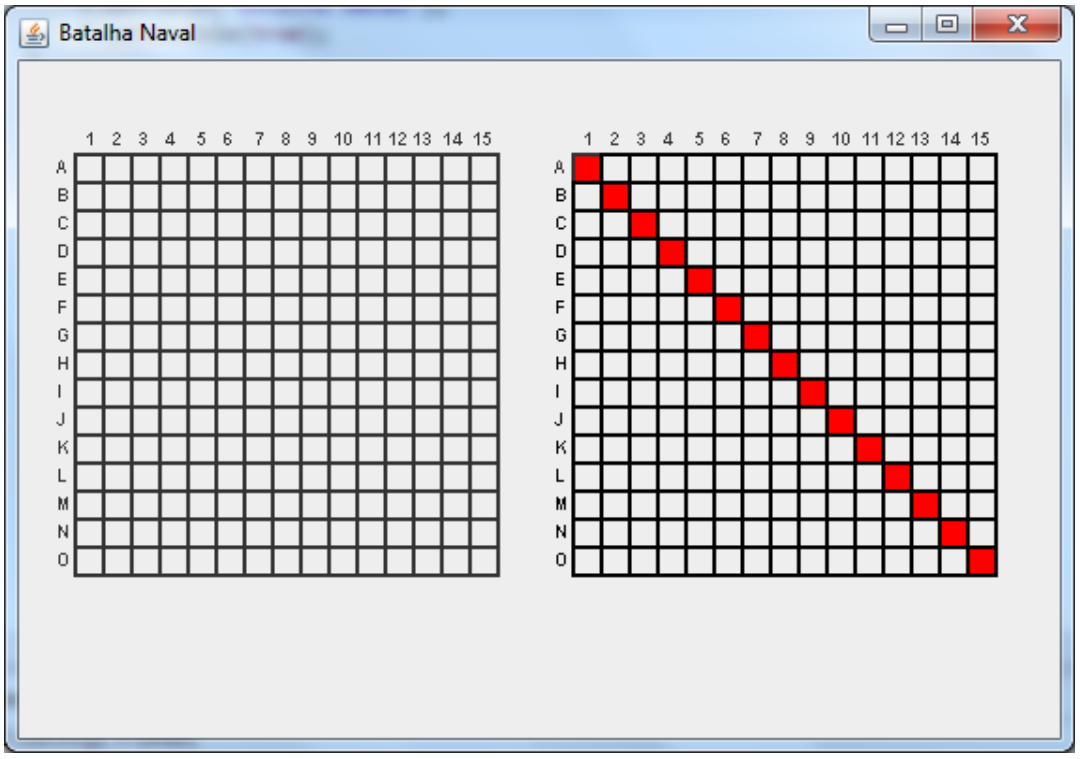
O objetivo deste trabalho é construir um programa que permita que duas pessoas joguem, em um único computador, partidas de Batalha Naval.

Descrição

1. Interface Gráfica

A interface gráfica do jogo deve usar elementos de Java Swing e Java2D. Os tabuleiros onde serão inseridas as embarcações (armas), por exemplo, devem ser construídos, **obrigatoriamente, por meio de objetos como Rectangle2D.Double e de métodos como fill(), draw() e drawstring(), encontrados na API Java 2D**.

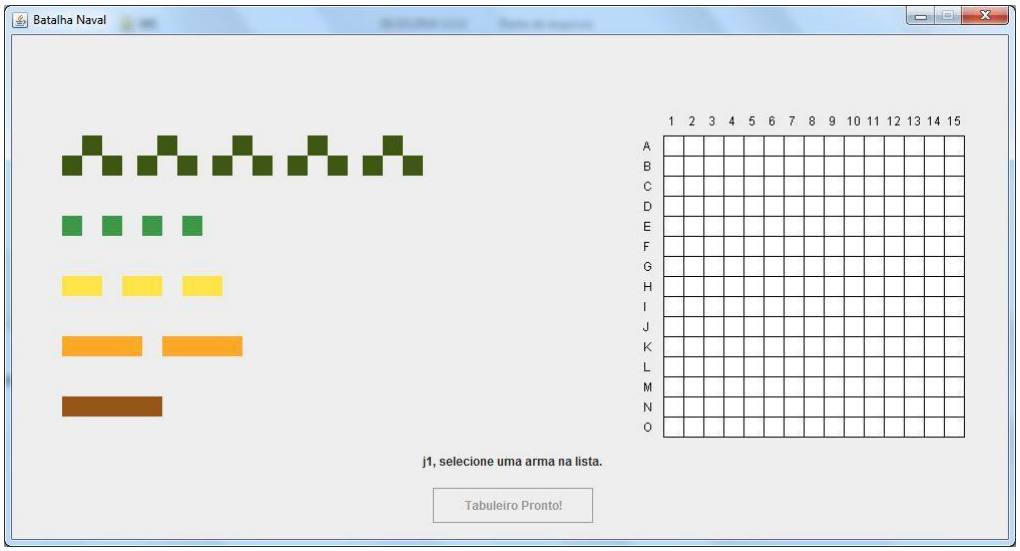
Use a figura abaixo como referência para a construção do seu tabuleiro.



Nenhum componente Java Swing, exceto um JPanel, poderá ser usado para facilitar o desenho dos tabuleiros e o tratamento de um clique de mouse sobre um tabuleiro. O

tratador de eventos de mouse terá de transformar as coordenadas cartesianas, obtidas do objeto de evento, em coordenadas matriciais (linha e coluna), que serão tratadas pelos componentes de regras do jogo de Batalha Naval.

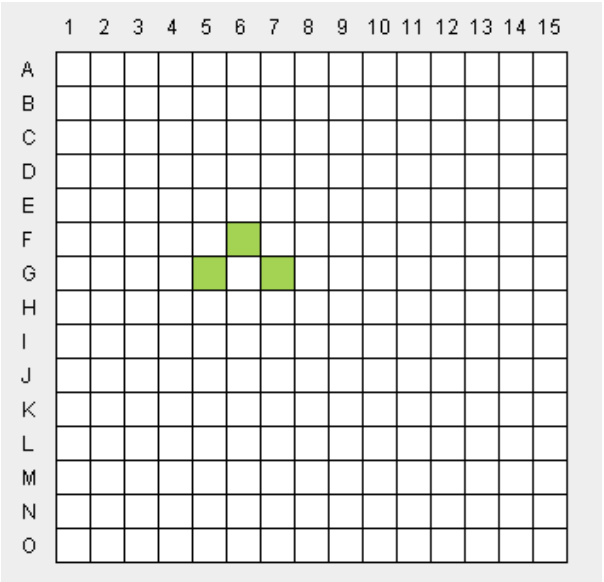
O posicionamento das armas no tabuleiro deve ser feito dividindo-se uma janela em duas partes: a primeira contendo as armas e a segunda a matriz de posicionamento.



O mecanismo de posicionamento deve seguir o seguinte padrão:

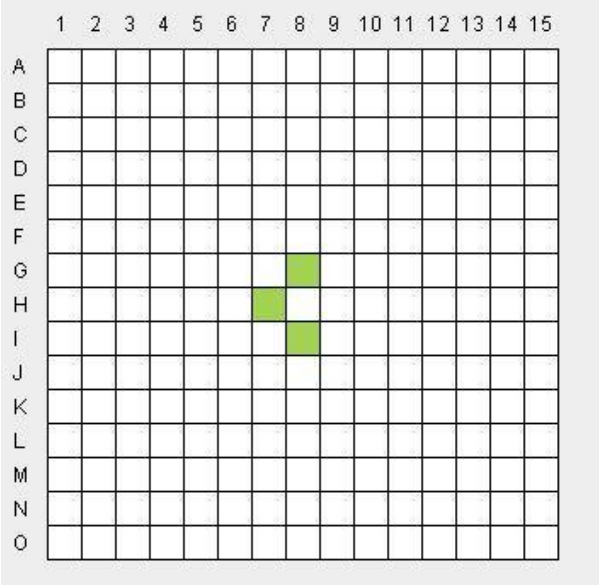
- Seleciona-se uma arma executando-se um clique sobre a mesma com o botão esquerdo do mouse (uma mudança de cor deve indicar a seleção).

- Seleciona-se um quadrado da matriz em que a arma deve começar a ser posicionada (clique sobre o quadrado com o botão esquerdo do mouse – vide figura a seguir).



- A arma pode ser deslocada fazendo-se um clique com o botão esquerdo mouse sobre outra posição qualquer da matriz.

- Podem-se executar giros de 90º, no sentido de sua escolha, pressionando-se o botão **direito** do mouse (vide figura a seguir).

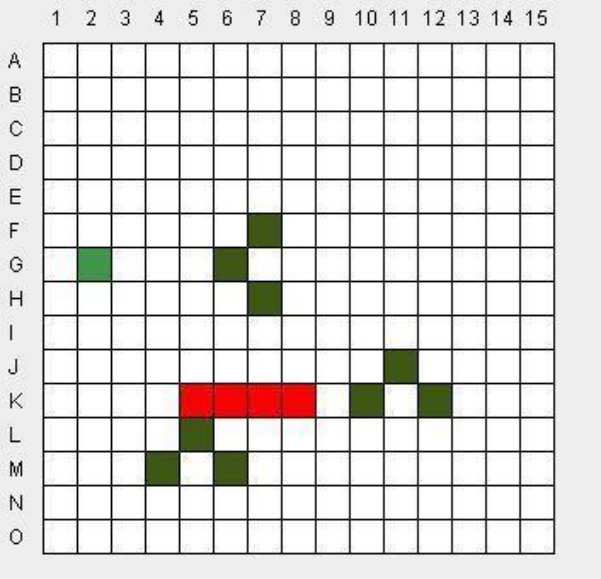


- O posicionamento definitivo é confirmado pressionando-se a tecla **Esc**.

- Pode-se reposicionar uma arma executando-se um clique com o botão esquerdo do mouse sobre qualquer um de seus quadrados. Após a seleção, o reposicionamento deve seguir os mesmos passos descritos acima.

- Caso uma arma seja posicionada incorretamente (por exemplo, compartilhando um lado de um quadrado com outra arma), ela deve ser exibida na cor vermelha (vide figura a seguir). Nessa condição o jogador poderá deslocá-la para outra posição, realizando um novo clique em outro quadrado, ou cancelar o posicionamento, pressionado a tecla **Esc**.

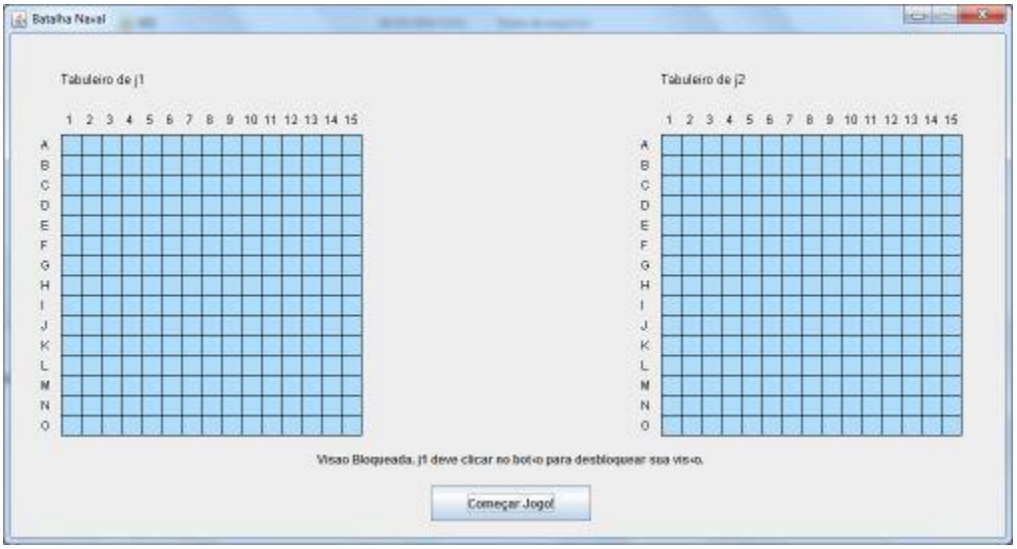
- A tela **Esc** também pode ser usada por um jogador para cancelar o posicionamento de uma arma que ainda não foi inserida na matriz (antes do jogador fazer um clique com o botão esquerdo do mouse sobre um quadrado da matriz).



Após todas as armas terem sido posicionadas deve-se habilitar um botão (JButton) para que o primeiro jogador passe a vez para o segundo jogador realizar o posicionamento de suas armas. As armas posicionadas pelo primeiro jogador não poderão mais ser reposicionadas.

2. Realização de Ataques

Após o segundo jogador confirmar o posicionamento de suas armas, o jogo deve passar para a etapa de ataques. Nesse momento o 1º jogador deverá visualizar um tabuleiro semelhante ao exibido pela figura a seguir.



Os tiros devem ser executados por meio de cliques com o mouse sobre o tabuleiro do jogador dois (tabuleiro da direita). Tiros na água devem ser mostrados pintando-se o quadrado com uma cor diferente da cor de fundo. Tiros que afundem uma parte de uma embarcação devem ser mostrados pintando-se o quadrado correspondente com uma 3ª cor. O afundamento de uma embarcação deve ser registrado pintando-se todos os quadrados dessa embarcação com uma 4ª cor.

O resultado de uma jogada deve ser informado por meio de um texto exibido em um **JLabel**. Use cores diferentes das usadas no resto do painel para que as mensagens sejam destacadas. Não use caixas de diálogos para informar os resultados, pois isso irá atrapalhar o andamento do jogo.

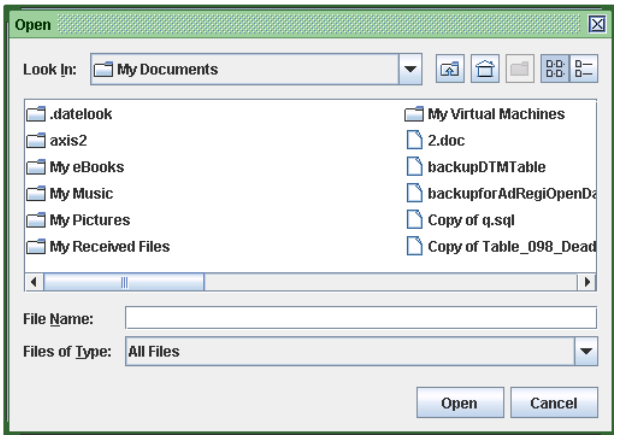
Após um jogador realizar seu terceiro tiro, o programa deverá informar que o outro jogador passará para o ataque. Use um componente gráfico, como um **JButton**, para que o jogador peça ao programa para mostrar a sua visão do jogo após ele assumir a posição de atacante. Não se esqueça de ocultar o tabuleiro do jogador que passou para a posição de atacado.

Quando um jogador afundar todas as armas de seu adversário, abra uma **Dialog Box** informando que ele venceu a partida. Limpe os tabuleiros, mas não feche o programa. Permita que uma nova partida possa ser iniciada sem que o programa tenha de ser encerrado.

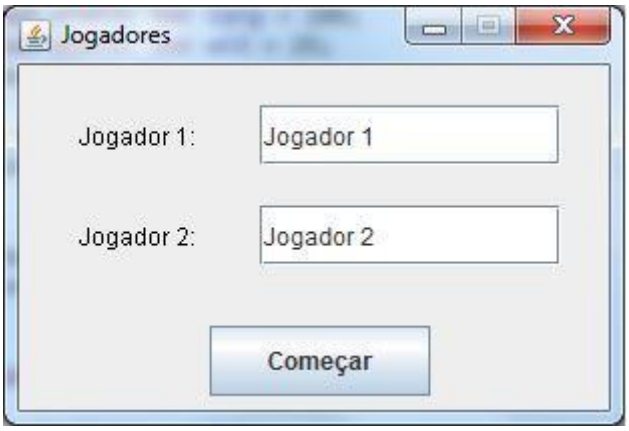
3. Início do Jogo

A primeira janela que será exibida pelo programa deverá permitir que os jogadores decidam se irão iniciar uma nova partida ou se irão continuar a disputar uma partida interrompida, que foi salva em um arquivo texto. Essa escolha pode ser feita por meio de **JButtons** ou itens de menu (**JMenu** e **JMenuItem**). Caso uma nova partida seja iniciada, será necessário definir os nomes dos participantes. Isso será exemplificado mais adiante.

É **OBRIGATÓRIO** que os jogadores possam escolher, por meio de um **JFileChooser**, o arquivo que desejam carregar (vide figura a seguir).



Caso uma nova disputa seja iniciada deve-se exibir um diálogo para que os jogadores se identifiquem. Esse diálogo deve tomar por base a figura a seguir.



4. Salvamento e Recarga de um Jogo

Após ambos os jogadores terem posicionado suas armas, deve ser possível salvar um jogo para recomeço em um momento posterior. O salvamento e a recarga devem ser feitos a partir de um item de menu (use os componentes JMenuBar, JMenu e JMenuItem), que, por sua vez, irá abrir um diálogo para a escolha de arquivo (JFileChooser). A opção de recarga deve estar disponível até o momento em que o 1º jogador posicionar sua primeira arma. Após isso, a única possibilidade de recomeçar um jogo será encerrar o jogo atual e começar um novo.

O estado de um jogo deverá ser gravado em um arquivo texto, cujo formato será definido em um momento posterior. Ele poderá residir em qualquer pasta do sistema de arquivos.

5. Regras do Jogo

As regras do jogo de Batalha Naval foram empacotadas no mesmo arquivo ZIP em que se encontra este enunciado.

Design e Implementação

O trabalho em questão terá de ser desenvolvido com a **linguagem Java** e a **última versão** da ferramenta **Eclipse**. **Trabalhos desenvolvidos com outras ferramentas não serão avaliados e seus participantes receberão nota ZERO no G2**.

Na avaliação do trabalho será levada em consideração a aplicação correta das técnicas de design e programação vistas durante o curso. Isso inclui a observação dos critérios de acoplamento e coesão, **a organização do aplicativo em pacotes** e a utilização obrigatória dos seguintes Design Patterns:

- Observer
- Façade
- Singleton

Todas as coleções que forem usadas no programa devem ser implementadas por meio de classes pertencentes ao framework de coleções de Java.