



Introdução

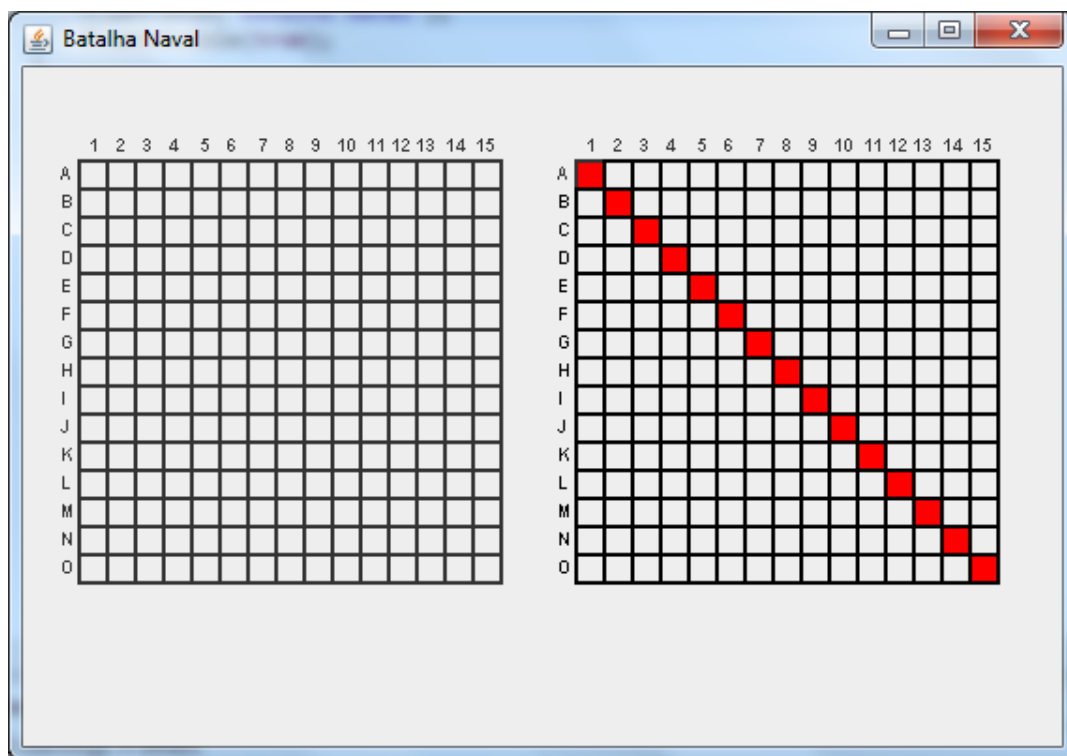
O objetivo deste trabalho é construir um programa que permita que duas pessoas joguem, em um único computador, partidas de Batalha Naval.

Descrição

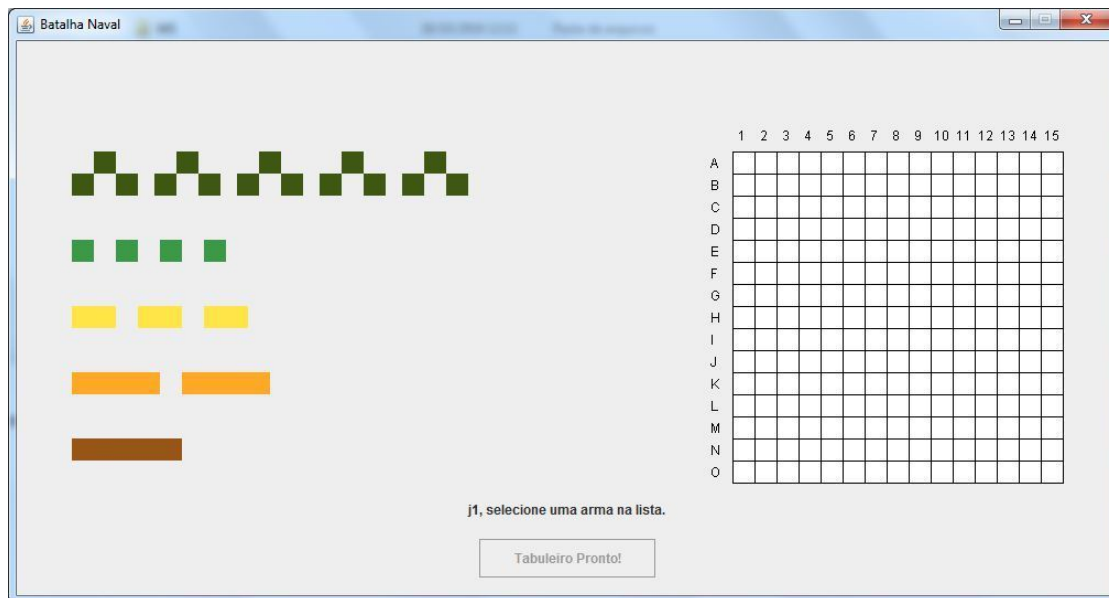
1. Interface Gráfica

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:

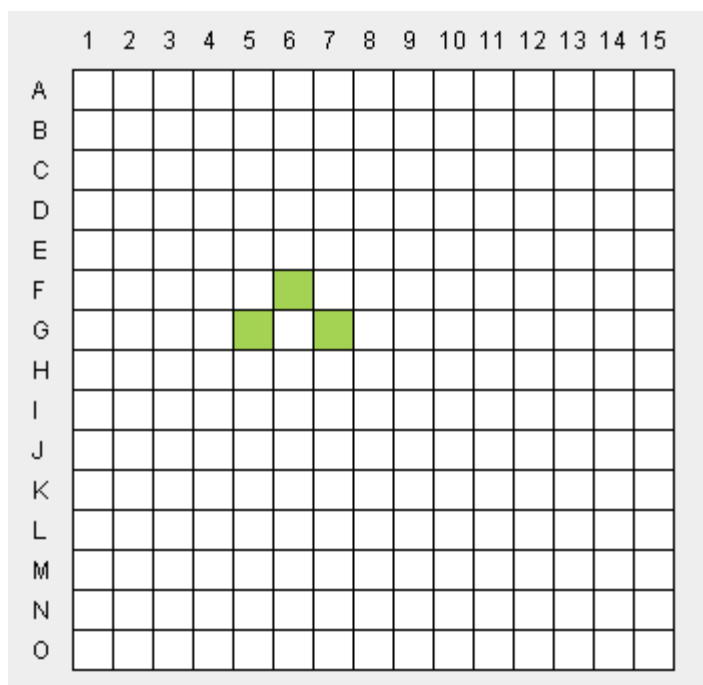


O posicionamento das armas no tabuleiro deve ser feito dividindo-se uma janela em duas partes: a primeira contendo as armas e a segunda a matriz de posicionamento.

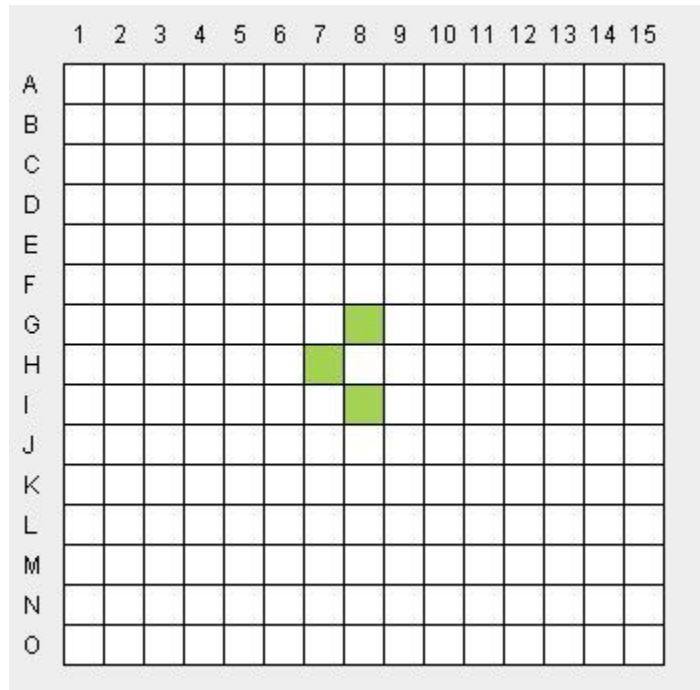


O mecanismo de posicionamento deve seguir o seguinte padrão:

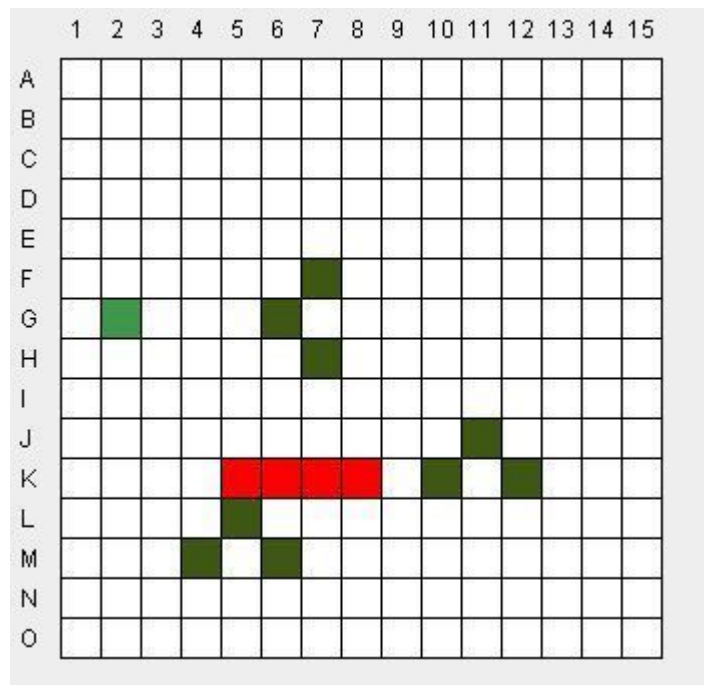
- Seleciona-se uma arma executando-se um clique sobre a mesma com o botão esquerdo do mouse (uma mudança de cor deve indicar a seleção);
- Seleciona-se um quadrado da matriz em que a arma deve começar a ser posicionada (clique sobre o quadrado com o botão esquerdo do mouse);



- 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 90° no sentido anti-horário pressionando-se o botão direito do mouse.



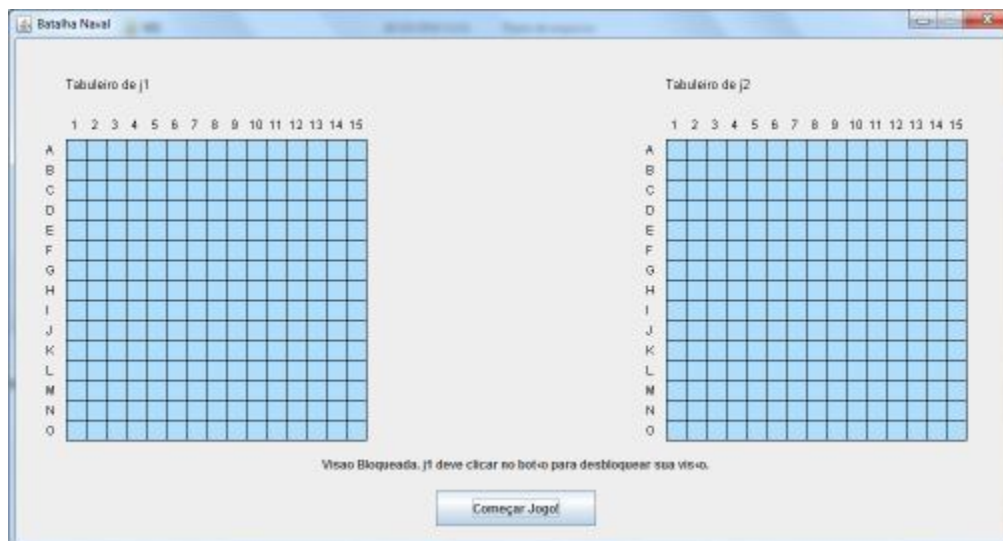
- O posicionamento definitivo é confirmado pressionando-se a tecla Esc;
- Pode-se reposicionar uma arma executando-se um clique com o botão esquerdo do mouse sobre qualquer um de seus quadrados. Após a seleção, o reposicionamento deve seguir os mesmos passos descritos acima;
- Caso uma arma seja posicionada incorretamente (por exemplo, compartilhando um lado de um quadrado com outra arma), ela deve ser exibida na cor vermelha. 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.



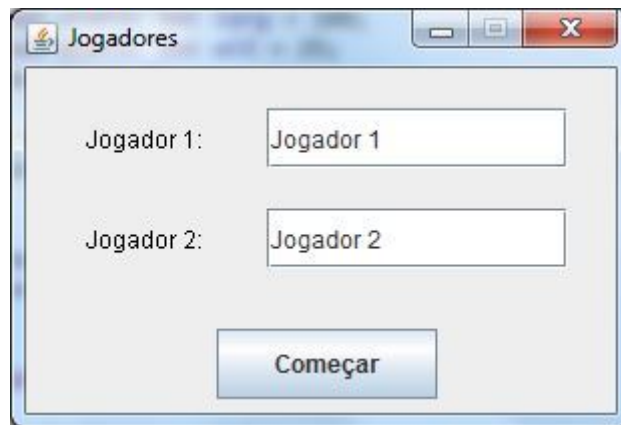
O resultado das jogadas devem ser informados por meio de um **JLabel**. Use cores diferentes das usadas no resto do painel para que as mensagens sejam destacadas. Não use caixas de diálogos para informar os resultados, pois isso irá atrapalhar o andamento do jogo.

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

Antes de um jogo ser iniciado 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 e 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 arquivo contendo o estado deve ser um arquivo texto, cujo formato será definido em um momento posterior. Ele poderá ser salvo ou aberto a partir de 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.

O Processo de Desenvolvimento

Este trabalho terá de ser desenvolvido com a **linguagem Java** e a **última versão** da IDE **Eclipse**. **Não examinarei os trabalhos desenvolvidos com outras ferramentas.**

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 o uso, obrigatório, dos seguintes Design Patterns:

- Observer;
- Facade;
- Singleton.

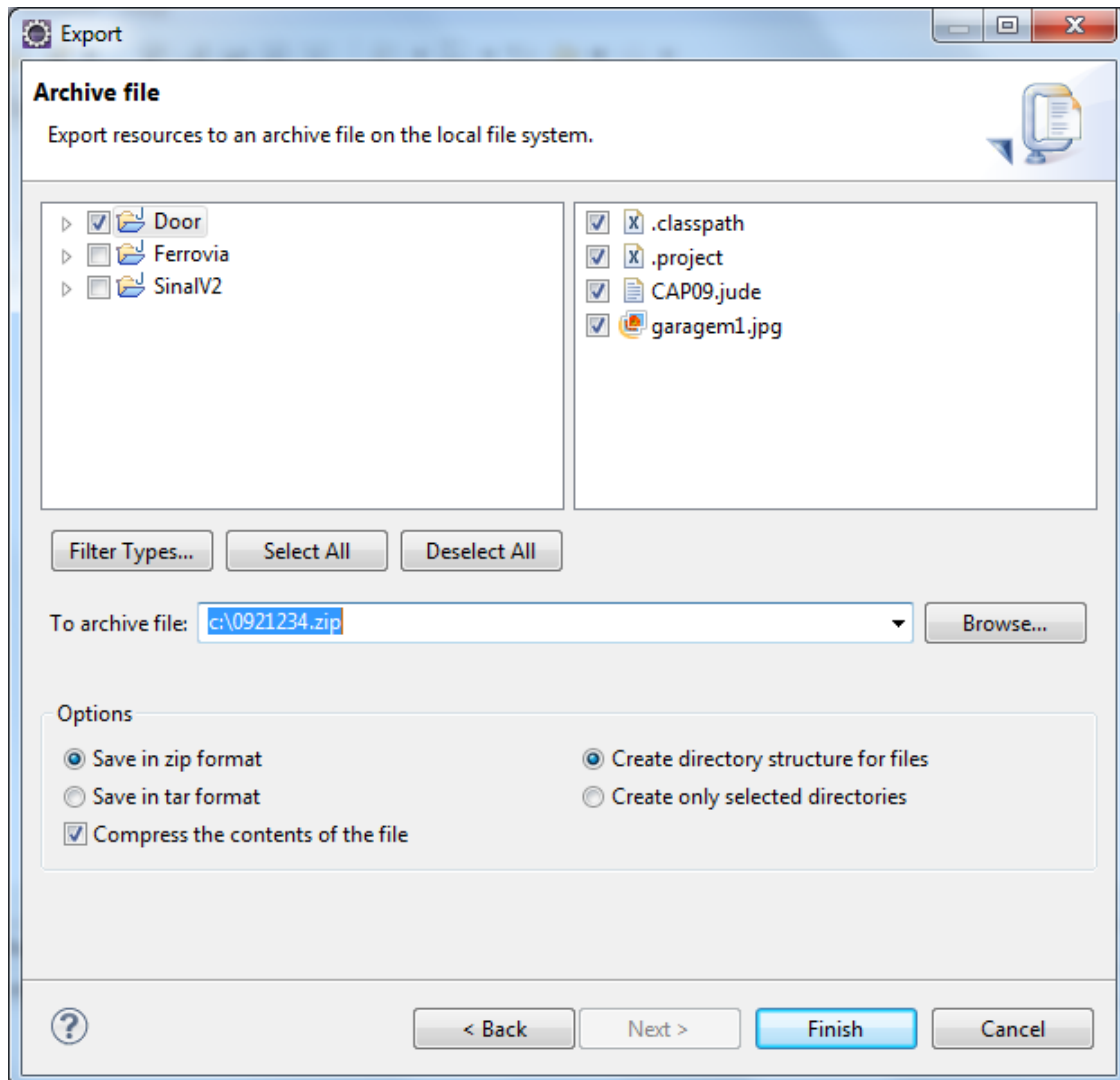
Artefatos a Serem Entregues

O projeto Eclipse com o código fonte Java deve ser exportado para um arquivo **ZIP** e enviado para o professor por meio do Ambiente de Aprendizagem On-line da PUC-Rio.

O nome do projeto **TEM** de ser a matrícula de um dos membros do grupo. Além disso, o nome do arquivo **ZIP** também tem de ser o número da matrícula.

Não criem projetos com nomes como, por exemplo, Batalha Naval, pois instalarei todos os projetos em um mesmo workspace. Logo, não pode haver dois projetos com o mesmo nome. O uso da matrícula como nome do projeto garante a unicidade de nomes.

Exporte o projeto para um arquivo **ZIP** como mostra a figura a seguir.



Iterações – Proposta de Organização das Tarefas

1ª Iteração

Data de término: 13/11/2016

Requisitos a serem implementados: Exibição do diálogo de identificação dos jogadores, exibição das matrizes de posicionamento, tratamento e identificação (definição das coordenadas da casa clicada) do clique do mouse e alternância dos tabuleiros dos dois jogadores.

2ª Iteração

Data de término: 27/11/2016

Requisitos a serem implementados: Posicionamento das armas, de ambos os jogadores, na matriz de posicionamento e transição para a fase de ataques.

3ª Iteração

Data de término: 04/12/2016

Requisitos a serem implementados: Realização dos ataques, com a marcação dos tiros, identificação das partes afundadas e dos tiros que caíram na água. Alternância de atacante para atacado e definição do vencedor.

4ª Iteração (última)

Data de término: 11/12/2016 (data de entrega)

Requisitos a serem implementados: salvamento e recarga de jogo salvo anteriormente.

Relatórios de Acompanhamento

Ao término de cada iteração cada participante terá de enviar um relatório descrevendo o andamento dos trabalhos. Os relatórios devem ser entregues por meio do Ambiente de Aprendizagem On-line da PUC-Rio. Eles devem ser individuais e seguir um modelo que será publicado posteriormente.

Cada componente do grupo deverá informar o que ele fez desde o término da última iteração e o que ele planeja fazer até o término da próxima. O relatório deve enumerar as tarefas que foram planejadas, mas não foram completadas, além de citar os motivos que levaram à não realização dessas tarefas.

O aluno que não entregar o relatório até a data e a hora definidas (**23h59 da data de término de uma iteração**) para o término de uma iteração será punido com a perda de **1,0 ponto** na nota final do trabalho.

Todos os grupos irão apresentar uma versão parcial do trabalho ao longo do processo de desenvolvimento. As datas serão definidas pelo professor, de forma aleatória, e serão informadas aos alunos com pelos 48h de antecedência. Durante a apresentação de uma versão parcial o professor poderá avaliar se o trabalho realizado até aquele momento corresponde ao que foi informado nos relatórios de acompanhamento.

Datas de Entrega e de Apresentações

1. A versão resultante da última iteração deverá ser entregue por meio do Ambiente de Aprendizagem On-line da PUC-Rio Isso terá de ser feito até as **23h59min** do dia **11/12/2016**. **Não haverá tolerância na entrega da versão final. Quem não entregá-la dentro do prazo definido receberá nota ZERO no G2.**
2. As apresentações ocorrerão nos dias **12, 13, 14, 15 e 16/12/2016**. **O aluno que faltar à sua apresentação receberá nota ZERO no G2.** A ordem e os horários das apresentações serão acertados com a turma na semana anterior às apresentações.

Regras Adicionais

O trabalho terá de ser feito em dupla. Trabalhos feitos por mais de dois alunos serão desconsiderados e **TODOS receberão grau ZERO.**

É proibido o compartilhamento de código entre os alunos. Caso fique evidente que um aluno copiou parte do código de outro trabalho, todos os alunos envolvidos receberão grau **ZERO**. Posteriormente o caso será levado ao Conselho Disciplinar do CTC.