Batalha Naval – versão 1.1

EP 1 – Entrega: 05/10/2008

Computação Móvel – MAC5743/MAC0463

Segundo Semestre de 2008

Prof. Alfredo Goldman (*gold at ime.usp.br*)

1. Introdução

Nesse exercício vocês vão implementar um jogo de Batalha Naval para celular, usando a tecnologia J2ME, explicada na seção 5. O jogo terá dois modos principais, o local e o conectado. Dessa forma, o usuário pode jogar contra uma máquina ou contra outra pessoa. Vejam na seção 3 quais as funcionalidades que o jogo deve ter, e na seção 4 alguns bônus que vocês podem implementar para melhorar a nota!

O trabalho **pode** e **deve** ser feito em duplas. A entrega deve ser feita pelo Paca, através de um arquivo **ZIP** ou **TAR.GZ** com o **nome da dupla**. Exemplos:

- `MarianaAlfredo.zip`
- `MarianaBravoAlfredoGoldman.tar.gz`

Nada de iniciais, por favor! Esse arquivo deve conter o **programa desenvolvido** (se for um projeto exportado do Eclipse, ajuda) e mais um **relatório** explicando detalhes sobre utilização e implementação do programa. Veja na seção 6 os critérios de avaliação, incluindo quais partes devem ser abordadas no relatório. Na seção 7, encontre mais informações sobre o jogo no modo conectado.

As dúvidas devem ser resolvidas através do fórum da disciplina no Paca.

2. Batalha Naval – O Jogo

O jogo de batalha naval é um clássico muito divertido em que o objetivo é afundar todos os navios do oponente antes que ele afunde os seus. É óbvio que você não sabe onde estão os navios do oponente – esse é o grande desafio!

Joga-se normalmente em um tabuleiro quadrado e quadriculado, que representa o mar. Antes de começar o jogo, cada jogador posiciona seus navios. Eles têm o mesmo grupo de navios a posicionar, e devem usar todos os navios sempre. Um jogador não pode ver o tabuleiro do outro, como já foi dito.

Depois disso, o jogo procede com cada jogador bombardeando uma posição no território do oponente, que informa se a bomba caiu na água ou se atingiu algum navio. Se um navio for atingido por completo, ele afunda, e o jogador é avisado disso também.

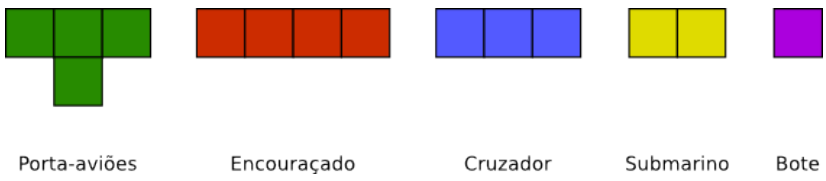

Figura 1: Barcos disponíveis

Na versão para celular que vocês devem desenvolver, existem cinco tipos de navios. Eles são o porta-aviões, o encouraçado, o cruzador e o bote, e suas formas podem ser vistas na Figura 1. O jogador dispõe de 1 porta-aviões, 1 encouraçado, 2 cruzadores, 3 submarinos e 4 botes para colocar num tabuleiro $10x10$. Na Figura 2 pode-se ver um exemplo de posicionamento das embarcações.

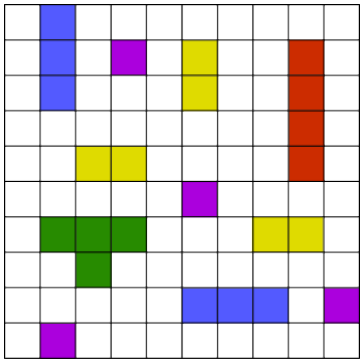

Figura 2: Exemplo de disposição dos navios

Observe que os navios nunca podem ser posicionados lado a lado (com um lado em comum), de forma a se encostar. Os navios podem, no entando, se encostar na diagonal (apenas um ponto em comum). Na Figura 3, o cruzador e o submarino estão numa posição proibida, enquanto o porta-aviões e o bote estão numa posição permitida.

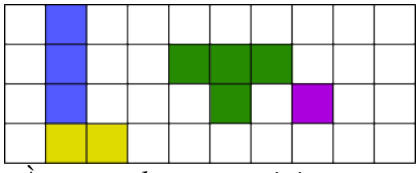

Figura 3: À esquerda, um posicionamento ilegal; à direita, um posicionamento válido.

3. O Jogo no Celular

Como já vimos, o jogo no celular terá dois modos: o local e o conectado. Quando o jogo é iniciado, o usuário pode escolher qual dos modos ele quer utilizar. Esses modos serão explicados a seguir, após uma discussão sobre apresentação e interação do programa.

Apresentação e Interação

A apresentação do tabuleiro ao jogador pode seguir um formato texto simples. Uma sugestão possível é mostrada na Figura 4. Repare que isso é apenas uma sugestão, e outras legendas ou até outras formas além de texto são permitidas para a apresentação. Também não é necessário ter símbolos diferentes para os diferentes tipos de barcos. A escolha é de vocês. Não precisa colocar a legenda no celular, desde que ela seja explicada no relatório.

Legenda:

`~`	Água
`P`	Porta-aviões
`E`	Encouraçado
`C`	Cruzador
`S`	Submarino
`B`	Bote
`?`	Desconhecido
`X`	Embarcação atingida
`.`	Água “atingida”

Pedaço de tabuleiro do jogador:

```
~~~PXP~~~~
~~~~P~~~~~
~X~~~~~S~~
~~.~~~~S~~
```

Pedaço de tabuleiro do oponente:

```
?~~???????
?XX????~??
??????????
????B~????
```

Figura 4: Exemplo de apresentação dos tabuleiros

A interação do usuário com o celular também fica a cargo de vocês, desde que seja

possível jogar como explicado a seguir. Tanto a apresentação quanto o modo de interagir com o programa serão avaliados e farão parte da sua nota. O programa deve ser fácil e simples de usar. Veja mais sobre os critérios de avaliação na seção 6.

Modo Local

Quando entra nesse modo, o jogo começa imediatamente, sendo dividido em duas etapas: a preparação e o jogo de fato.

Preparação do jogo

Durante a preparação do jogo, o usuário posiciona um a um os seus navios no tabuleiro 10*x*10. Ele deve poder girar os navios o quanto quiser, além de movê-los para qualquer posição.

Nessa etapa, o celular também deve escolher uma posição para as peças do jogador automático, de modo aleatório. Não vale escolher sempre o mesmo tabuleiro!

O jogo

O jogo local sempre começa com o usuário jogando, e consiste de alternar as seguintes ações:

1. Aparece na tela o território do inimigo, escondendo as posições desconhecidas.
2. O usuário escolhe uma posição para atingir. O celular verifica e mostra o resultado:
    1. Se for mar, revela mar na posição (~ no exemplo);
    2. Se for um barco que não afundou, mostra um símbolo de posição atingida (X no exemplo);
    3. Se for um barco que afundou, troca os X no barco por símbolos representando o tipo de barco afundado (como o barco B do exemplo).
3. Depois de mostrar o resultado, a vez passa ao jogador automático. O usuário vê em sua tela o seu próprio tabuleiro.
4. O jogador automático escolhe uma posição para atingir (por exemplo aleatória), e o usuário vê em seu território o quê o adversário descobriu:
    1. Se descobriu água, ele vê um símbolo de água “atingida” (. no exemplo);
    2. Se atingiu um navio mas não afundou, ele vê um símbolo que um dos seus navios foi atingido (por exemplo, X);
    3. Se afundou um navio, o usuário é informado de alguma forma, por exemplo usando símbolos diferentes ou através de uma mensagem, ou removendo o barco de seu tabuleiro.
5. Volta-se ao passo 1.

Repare que nenhum dos jogadores pode atingir uma posição que já tenha atingido! Além de ser uma jogada nada inteligente, facilitaria muito o trabalho do jogador automático pra vocês!

Por fim, quando todos os barcos de algum dos jogadores forem afundados, o programa informa ao usuário quem foi o vencedor (quem ainda tem barcos sem afundar).

Modo Conectado

Se o jogador decide que quer jogar contra outra pessoa (ao invés de um programa), o celular deve se conectar ao servidor de Batalha Naval e trocar mensagens com ele para que o usuário possa jogar. Essa troca de mensagens deve seguir um protocolo de comunicação rígido, que pode ser lido na seção 7 deste enunciado.

A seguir, uma breve descrição de como funcionará a interação com o servidor.

Conexão no servidor

Para estabelecer um jogo entre dois celulares, o servidor aguarda que duas pessoas se conectem e, assim que isso acontecer, inicia um jogo entre os dois. Logo, quando um celular se conecta ao servidor, existem duas possibilidades: já tem alguém esperando, e os dois iniciam imediatamente um jogo; ou não tem ninguém esperando, e o celular terá que esperar até que outra pessoa se conecte. Durante esse período de espera, o celular deve "pingar" o servidor de tempos em tempos para permanecer esperando.

Preparação do jogo

Uma vez que um jogo entre dois celulares começa, ambos passam à etapa de preparação do jogo. Ou seja, os usuários configuram seus tabuleiros exatamente da mesma maneira que com o jogo local. Depois, cada celular envia seu tabuleiro para o servidor e o jogador que estava a mais tempo esperando começa o jogo. O primeiro jogador a enviar o tabuleiro deve "pingar" o servidor até obter a resposta de que o jogo vai começar.

O jogo

O jogo, da perspectiva do jogador, ocorre exatamente da mesma maneira que um jogo local. Por baixo dos panos, no entanto, o celular faz toda a comunicação com o servidor.

Quando o usuário faz uma jogada, o celular a envia ao servidor, recebendo de volta o efeito que essa jogada provocou, para mostrar ao usuário. Quando o usuário está esperando a jogada de seu oponente, o celular pergunta de tempos em tempos ao servidor qual foi essa jogada, até obter uma resposta. Quando vier a resposta, ele mostra ao usuário e volta a vez para ele.

Isso se repete até o servidor declarar que algum dos usuários venceu.

4. Bônus

Existem duas funcionalidades bônus que podem ser implementadas para aumentar sua nota:

- **Jogador(es) automático(s) extra(s):**

  Além do jogador que é obrigatório implementar, você pode fazer outros tipos de jogadores, tornando a experiência mais divertida pro usuário que gosta de jogar sozinho! Assim, ao iniciar um jogo local, o usuário pode escolher contra qual "robô" ele quer jogar. Note que o jogador não precisa ser mais ou menos inteligente que o "padrão", basta que eles sigam estratégias diferentes.

- **Formato de jogo fácil:**

  Algumas pessoas gostam de jogar mais vezes jogos mais rápidos. Para isso, você pode oferecer, além do formato tradicional de jogo, um formato fácil. Supostamente, esse formato é mais rápido de terminar. Ele é jogado num tabuleiro 6*x*6 e os barcos disponíveis são 1 encouraçado, 1 cruzador e 2 submarinos. Essa opção, por motivos de compatibilidade, também só está disponível para o jogo local.

5. J2ME

J2ME (Java 2 Platform, Micro Edition), agora conhecido como Sun Java Wireless Toolkit for CLDC, é a adaptação do ambiente de execução Java para o uso em dispositivos que possuem recursos limitados. Em geral, esses dispositivos possuem pouca memória, telas pequenas, métodos alternativos de entrada de dados e pouco poder de processamento.

Para o desenvolvedor Java, aprender J2ME não é difícil: uma vez que conhecida a nova terminologia, o programador só precisa se acostumar com uma nova API e aprender a desenvolver em um ambiente limitado (alguns aparelhos celulares só possuem 30k de memória).

Para começar a aprender a tecnologia, sugiro os seguintes links:

- http://developers.sun.com/mobility/midp/articles/wtoolkit/ - tutorial básico

- http://developers.sun.com/mobility/reference/codesamples/ - mais exemplos
- http://developers.sun.com/mobility/reference/index.jsp – links para mais recursos
- http://j2meunit.sourceforge.net/ - testes unitários ao estilo JUnit

Note que não precisaremos de persistência para esse EP! Além disso, testes unitários para as classes do programa são bem-vindos!

Como ambiente de programação, sugiro plugins para duas das plataformas mais usadas para desenvolvimento Java:

- Eclipse: http://eclipseme.org/docs/installation.html
- NetBeans: http://www.netbeans.org/kb/articles/mobility.html

Usem e abusem do fórum da disciplina no Paca para tirar dúvidas, responder dúvidas e discutir em geral coisas sobre a tecnologia e sobre o EP. Não deixe de perguntar se tiver dúvidas ou dificuldades!!

6. Correção

Os trabalhos serão corrigidos no Linux, utlizando os seguintes programas/versões:

- Eclispse 3.4 com EclipseME 1.7.9
- Sun Java Wireless Toolkit 2.5.2 for CLDC
- Java SE 1.6
- J2ME Unit 1.1.1

Os trabalhos serão avaliados nos seguintes aspectos:

- Funcionamento: possibilidade de jogar nos dois modos, como descrito
- Interface e apresentação: facilidade, limpeza, explicação no relatório
- Código: clareza, organização, presença de testes, explicações no relatório

7. Jogo conectado – protocolo de comunicação

Para jogar com outros celulares, o seu programa deverá se comunicar por HTTP com o servidor de Batalha Naval.

Para todos os comandos, a requisição deve ser feita utilizando o método `POST`. Para facilitar a troca de informações entre cliente e servidor, vamos transmitir os dados no corpo do `POST`, usando os fluxos `DataInputStream` e `DataOutputStream` do Java, dessa forma os tipos primitivos e `String` podem ser facilmente enviados/recebidos.

A seguir estão definidos a seqüência e o tipo do valor para cada operação no servidor. Para o envio de informações do tipo `String` use o método `writeUTF`. Para receber os dados, use os métodos `readUTF` para `String` e `readInt` para `int`.

Uma posição do tabuleiro será identificada por duas coordenadas, a primeira para sua linha e a segunda para sua coluna. Suas linhas são identificadas pelas letras de A até J, e suas colunas pelos números de 1 a 10. Sendo assim, a posição mais à esquerda na primeira linha é identificada pela String “A1”, e a posição mais à direita na última linha pela String “J10”.

Vamos relembrar a seguir as etapas do modo conectado, e ver ao mesmo tempo as mensagens que pertencem a cada etapa.

Conexão no servidor

Nessa etapa, o celular se conecta ao servidor e espera (ou não) outro jogador para iniciar um jogo. Nessa etapa, quando algum erro ocorrer o celular deve se conectar novamente.

Existem duas mensagens:

Conectar

Entra no servidor. Recebe um `id` que identifica esse celular para o servidor, e deve ser usado em todas as mensagens subsequentes.

- **Pedido**:
  String “op=conecta”
- **Respostas**:
  - *Jogador deve esperar que outro jogador se conecte*
    String “OK espera”
    int “#id”
  - *Jogador deve passar para a etapa de preparação*
    String “OK joga”
    int “#id”
  - *Houve algum erro fatal*
    String “ABORTA mensagem de erro”

Ping

Enquanto receber aviso de esperar, o celular deve enviar uma mensagem de `ping` ao servidor. O intervalo entre as mensagens deve ser de no máximo 1 minuto, caso contrário o celular perderá a conexão com o servidor.

- **Pedido**:
  String “op=ping”
  String “id=#id”
- **Respostas**:
  - *Jogador deve esperar que outro jogador se conecte*
    String “OK espera”
  - *Jogador deve passar para a etapa de preparação*
    String “OK joga”
  - *Houve algum erro fatal*
    String “ ABORTA mensagem de erro”

Preparação do jogo

Nessa etapa, o usuário monta seu tabuleiro e o celular o envia para o servidor. Alguns erros são leves e só precisam que a mensagem seja reenviada corretamente, outros são fatais e precisam que o celular reconecte ao servidor.

Envio de tabuleiro

Envia o tabuleiro completo montado pelo usuário para o servidor. Para cada tipo de barco, segue uma descrição do formato a ser utilizado na mensagem:

Tipo de barco	Formato	Observações
Bote	bote=#coord	
Submarino	submarino=#coord,#coord	
Cruzador	cruzador=#coord,#coord,#coord	As coordenadas devem ser fornecidas na seqüência da linha ou da coluna, sem pular.
Encouraçado	encouracado=#coord,#coord,#coord,#coord	
Porta-aviões	portaavioes=#coord,#coord,#coord,#coord	As três primeiras coordenadas devem ser

		fornecidas na seqüência da linha ou da coluna, e a última deve ser a coordenada que forma “a perna do T”.

O formato das coordenadas foi descrito no início desta seção. A seguir a mensagem a ser usada para envio do tabuleiro:

- **Pedido**:
  String “op=tabuleiro”
  String “id=#id”
  Essas duas linhas são seguidas por 11 linhas, uma descrevendo cada barco, em qualquer ordem:
  String “#formato de barco” ...
- **Respostas**:
  - *Jogador deve esperar que o oponente envie seu tabuleiro*
    String “OK espera”
  - *Jogo vai começar. Jogador será o primeiro a jogar.*
    String “OK joga”
    String “comeca”
  - *Jogo vai começar. Jogador não será o primeiro a jogar.*
    String “OK joga”
    String “espera”
  - *Houve algum erro leve*
    String “ERRO mensagem de erro”
  - *Houve algum erro fatal*
    String “ABORTA mensagem de erro”

Ping

Depois que enviar seu tabuleiro, se receber a mensagem de espera, o celular deverá enviar pings ao servidor, com no máximo 1 minuto de intervalo, até receber a mensagem de que o jogo vai começar.

- **Pedido**:
  String “op=ping”
  String “id=#id”
- **Respostas**:
  - *Jogador deve esperar que o oponente envie seu tabuleiro*
    String “OK espera”
  - *Jogo vai começar. Jogador será o primeiro a jogar.*
    String “OK joga”
    String “comeca”
  - *Jogo vai começar. Jogador não será o primeiro a jogar.*
    String “OK joga”
    String “espera”
  - *Houve algum erro leve*

String “ERRO mensagem de erro”

- *Houve algum erro fatal*
  String “ABORTA mensagem de erro”

O jogo

Nessa etapa, os jogadores jogam! Eles devem alternar entre as mensagens a seguir até o fim do jogo.

Fazer uma jogada

Quando é a vez do jogador, ele deve fazer uma jogada.

- **Pedido**:
  String “op=joga”
  String “id=#id”
  String “posicao=#coord”
- **Respostas**:
  - *Jogada realizada com sucesso, bomba caiu na água*
    String “OK agua”
  - *Jogada realizada com sucesso, bomba atingiu mas não afundou algum barco*
    String “OK atingiu”
  - *Jogada realizada com sucesso, bomba afundou algum barco*
    String “OK afundou”
    String “#nomedobarcoafundado”
  - *Jogada realizada com sucesso, bomba afundou o último barco*
    String “OK venceu”
    String “#nomedobarcoafundado”
  - *Houve algum erro leve*
    String “ERRO mensagem de erro”
  - *Houve algum erro fatal*
    String “ABORTA mensagem de erro”

Espera o oponente fazer uma jogada

Quando é a vez do oponente, ele deve enviar pings para o servidor com intervalo máximo de 1 minuto para saber quando o oponente fez sua jogada.

- **Pedido**:
  String “op=recebe”
  String “id=#id”
- **Respostas**:
  - *Oponente ainda não jogou, jogador deve esperar mais*
    String “OK espera”
  - *Oponente jogou na posição informada e o jogo continua*
    String “OK continua”
    String “posicao=#coord”
  - *Oponente jogou na posição informada e jogo terminou, oponente é o vencedor*
    String “OK fim”
    String “posicao=#coord”

- *Houve algum erro leve*
  String “ERRO mensagem de erro”
- *Houve algum erro fatal*
  String “ABORTA mensagem de erro”

Divirtam-se!