

# Técnicas de Desenvolvimento de Software

Licenciatura em Engenharia Informática e de Computadores

Semestre de Verão de 2021/1022

## 1º Trabalho prático

A avaliação da componente prática da disciplina será realizada com base na aplicação *Battleship*, a ser desenvolvida ao longo do semestre. A aplicação viabiliza que dois jogadores, cada um no seu computador, joguem à batalha naval. O desenvolvimento da aplicação será realizado em duas fases. O presente documento descreve os requisitos da primeira fase (primeiro trabalho), para desenvolver a aplicação em modo consola, onde as jogadas são descritas textualmente e o tabuleiro é também apresentado textualmente, com o aspecto semelhante ao apresentado na figura.

Na segunda fase (segundo trabalho) será desenvolvida uma aplicação com interface gráfica onde as jogadas são realizadas com clicks do rato nas quadrículas da grelha. Importa sublinhar que se pretende maximizar a reutilização do código implementado na primeira fase, concretamente, o código relativo ao modelo do jogo, ou seja, o estado do tabuleiro, as jogadas já realizadas, e a lógica das regras do jogo, ou seja, tudo aquilo que não seja relativo à interface com o utilizador.

A aplicação em anexo ([Trabl.jar](#)) é uma demo da aplicação pretendida na primeira fase. Esta aplicação usa ficheiros de texto localizados no diretório de trabalho para armazenar o estado corrente dos jogos, em vez de uma base de dados remota, conforme requerido na implementação a ser produzida pelos alunos. O formato do conteúdo do ficheiro não é mandatório. Para usar esta demo devem executar duas instâncias usando o mesmo diretório de execução e o mesmo computador.

Para executar a demo, numa janela de comandos use **kotlin Trabl.jar** ou **java -jar Trabl.jar**

O jogo é realizado em duas etapas. Primeiro o jogador define os navios da sua frota usando os comandos `put`, `remove` e `grid`. Quando a frota estiver completa pode ser iniciado o jogo usando o comando `start`. Quando os dois jogadores iniciarem o jogo começa a etapa dos disparos usando os comandos `shot`, `refresh` e `grid`. O jogo termina quando um dos jogadores afunda toda a frota do adversário ou sai do jogo com o comando `exit`.

A aplicação suporta os seguintes comandos:

<b>put</b> <shipType> <position> <align>	Adiciona à frota um navio do tipo <shipType> a partir da posição <position> alinhado na vertical ou horizontal conforme <align>. <shipType> pode ser o prefixo do nome ou o número de quadrículas. <position> é o canto superior esquerdo indicado no formato <column><row>. <align> é V ou H.
<b>remove</b> <position>	Remove o navio da frota que tenha uma das posições ocupadas em <position>.
<b>grid</b>	Se o jogo ainda não foi iniciado apresenta a grelha com a frota já constituída até ao momento. Se o jogo já foi iniciado apresenta a grelha do próprio e a grelha do adversário com os disparos já realizados.
<b>start</b> <gameld>	Começa um jogo com o nome indicado. Caso seja o primeiro jogador a iniciar o jogo, o jogador terá que espera que o segundo jogador também inicie o jogo com o mesmo nome (ver comando refresh)

```
> shot b10
Miss
      A B C D E F G H I J      A B C D E F G H I J
+-----+ +-----+
1 | #  #  #  #  #  |  |      0      |
2 | #  *  #  #  #  |  |      X      |
3 |  *  #  #  #  #  |  |      X      |
4 | #  *  #  #  #  |  |      |
5 | #  0  #  #  #  |  |      |
6 |  #  #  #  #  #  |  |      |
7 | #  #  #  #  #  |  |      |
8 | #  #  #  #  #  |  |      |
9 | #  #  #  #  #  |  |      *      |
10 | #  #  #  #  #  |  |      0      |
+-----+ +-----+
Wait for other (use refresh command)
> |
```

<b>shot</b> <position>	Realiza um disparo na grelha do adversário na posição indicada, mostrando o resultado do disparo e apresenta automaticamente a grelha (comando grid).
<b>refresh</b>	Atualiza o estado do jogo depois para verificar se o adversário já realizou mais ações.
<b>exit</b>	Termina o jogo e a aplicação.

Cada símbolo apresentado na grelha pelo comando `grid`, representa o seguinte:

<b>#</b>	Quadrícula ocupada por navio da própria frota que ainda não foi atingido nessa posição.
<b>O</b>	Disparo que não acertou em nenhum navio.
<b>*</b>	Disparo que acertou num navio ainda não afundado (tipo não identificado).
<b>X</b>	Disparo que acertou num navio que foi afundado (tipo já identificado).

Os navios de cada frota não podem ficar encostados lado a lado nem canto com canto. O comando `put` deve verificar estas condições.

O primeiro jogador a ter iniciado o jogo será o primeiro a disparar. A vez de disparar só muda para o adversário quando o último disparo não acertar num navio.

Opcionalmente, podem ser implementadas as variantes do comando `put`:

<b>put</b> <shipType>	Adiciona à frota um navio do tipo <shipType> numa posição a alinhamento aleatórios .
<b>put all</b>	Adiciona os restantes navios da frota em posições e alinhamentos aleatórios

Para viabilizar que cada jogador faça as jogadas no seu computador, a informação armazenada do estado do jogo é armazenada num documento, cuja identificação é o nome do jogo iniciado por cada jogador, na base de dados documental acessível a partir das máquinas onde o jogo esteja a ser executado. O formato do documento deve ser o adequado à informação absolutamente necessária.

Bom trabalho!  
Pedro Pereira

ISEL, 5 de Abril de 2022