

RebelWars

Programação Orientada pelos Objectos

Enunciado do 1º Trabalho Prático, versão 1.4 – 2017-04-18

Contacto: carla.ferreira@fct.unl.pt

Alterações em relação à versão 1.0: Alterações de texto são apresentadas a **vermelho**.

Notas importantes:

Este enunciado diz respeito ao primeiro trabalho de POO1617.

Prazo de entrega: até às 23h55 do dia 30 de Abril de 2017.

Realização em grupos: grupos de 2 alunos.

Material a entregar

- **Moodle:** ficheiro **zip** submetido via Moodle contendo o **projecto eclipse completo** da aplicação desenvolvida, com código fonte devidamente comentado, o respectivo **Javadoc** e o **diagrama de classes e interfaces**. Por favor inclua o seu nome e número de aluno em todos os seus ficheiros fonte (*.java), em cláusulas @author. Compacte todos estes artefactos num ficheiro cujo nome tem de ser da forma "XXXXXX_YYYYY", com XXXXX < YYYYY, onde XXXXX e YYYYY representam os números dos membros do grupo (exemplo: 12345_54321.zip).
- **Mooshak:** submissão e aceitação do código fonte pelo sistema de avaliação automática Mooshak (pontuação > 0). Consulte na página da disciplina as regras de submissão de trabalhos ao Mooshak.

Recomendações: A boa documentação do seu código fonte será valorizada, tal como a utilização do melhor estilo de programação possível, para além, claro, do correcto funcionamento do projecto. Não se esqueça de comentar as declarações das classes e das interfaces com uma explicação do significado das mesmas.

1. Desenvolvimento de aplicação RebelWars

1.1. Descrição do problema

O objectivo deste trabalho é desenvolver um jogo de computador para ajudar os rebeldes a colectar armas para o exército da Aliança Rebelde. Os heróis do jogo são os rebeldes que estão a ser perseguidos pelos *stormtroopers* (vilões). Usaremos ao longo do enunciado os termos rebelde e herói para nos referirmos à mesma entidade. Os personagens deslocam-se num labirinto que é um espaço bidimensional. Os movimentos dos rebeldes são controlados pelo jogador, enquanto os *stormtroopers* obedecem cegamente a ordens superiores (são controlados pelo computador).

Os rebeldes deslocam-se no labirinto colectando armas que foram escondidas em vários locais do planeta Quesh e tentando não ser capturados pelos *stormtroopers*. O jogo termina quando os rebeldes colectaram todas as armas ou foram todos capturados pelos *stormtroopers*. Existem localizações que têm uma poção escondida (*adrenals*) que dá ao rebelde poderes especiais (mais à frente vamos ver quais).

Os personagens deslocam-se numa de 4 direcções: cima, baixo, esquerda e direita. Nenhum dos personagens atravessa paredes.

A aplicação tem uma funcionalidade para a leitura do labirinto. Após a leitura do labirinto é possível adicionar vários rebeldes. Os heróis podem ser colocados numa qualquer posição vazia do labirinto. O número máximo de heróis que podem ser adicionados ao jogo está limitado pelo número de posições livres do labirinto.

O labirinto é representado por um rectângulo com **l** linhas e **c** colunas. Pode assumir que o labirinto será sempre um rectângulo. O canto superior esquerdo tem a coordenada (1, 1) e o canto inferior direito a (**l**, **c**). Cada carácter destas linhas tem um significado:

- '#' – parede do labirinto;
- ' ' – posição vazia no labirinto;
- '.' – posição com uma arma;
- 'P' – posição com uma poção;
- 'B', 'W', 'O' – *stormtrooper* (black, white, orange);
- 'R' – posição com um rebelde;
- 'S' – posição com um rebelde após tomar a poção.

Quando o labirinto é lido, este inclui todos os caracteres apresentados acima excepto os dois últimos que representam os soldados rebeldes. Depois de ler o labirinto o jogador pode criar os heróis do jogo. O jogo pode ser iniciado se o labirinto já tiver sido lido e existir pelo menos um soldado rebelde, ou seja, a fase de *setup* já tiver sido completada. Após iniciar o jogo já não é possível adicionar mais rebeldes.

Estando o jogo iniciado o jogador tem que dar instruções de movimentação aos rebeldes. Se um soldado rebelde se cruzar com um *stormtrooper* é capturado. Se passar por uma posição com uma arma, a arma é colectada e são-lhe atribuídos 10 pontos. Quando a arma é colectada essa posição passa a estar vazia. Os *stormtroopers* não têm qualquer efeito sobre as posições com armas (porque as armas estão camufladas).

As personagens capturadas não voltam a ser mostradas no labirinto, mas deve ser possível consultar a sua informação.

A poção dá poderes especiais ao rebelde que a encontrar, mas não tem qualquer efeito nos *stormtroopers*. Após beber a poção, e durante 4 jogadas, esse soldado passa a ser representado pelo símbolo 'S', e pode capturar *stormtroopers*. Após a poção ser consumida, essa posição passa a estar vazia.

O labirinto seguinte tem 9 linhas e 27 colunas, 10 localizações onde se podem colocar rebeldes, 2 poções mágicas e 10 *stormtroopers* (3 de cor laranja, 3 de cor branca, 4 de cor preta):

```

.....OOO.....WWW.....
.....#
P#####   #####   #####P#
..#.....#.....#.....#
..###.....#.....#.....#
..#.....#.....#.....#
..#.....#####.....#.....#
.....
.....BBBB.....

```

Cada rebelde tem um nome único, o estado (activo, poção activa, capturado), a posição corrente no labirinto, os pontos obtidos e o trajecto que percorreu no labirinto. Cada *stormtrooper* tem um identificador único, o estado (activo ou capturado), uma cor, e os rebeldes capturados até altura. Para o labirinto acima os identificadores dos *stormtroopers* seriam: “ST-O-1”, “ST-O-2”, “ST-O-3”, “ST-W-4”, “ST-W-5”, “ST-W-6”, “ST-B-7”, “ST-B-8”, “ST-B-9”, “ST-B-10”. Ou seja, é seguida uma ordenação primeiro por linha e depois por coluna.

Não há limite ao número de rebeldes (*stormtroopers*) que um stormtrooper (*rebelde*) pode capturar.

Como já foi dito os *stormtroopers* são controlados pelo computador. Na prática cada tipo de *stormtrooper* faz repetidamente uma sequência de movimentos pré-definidos. **Em cada momento, o stormtrooper vai tentar seguir o movimento que estiver seleccionado na sua sequência, apenas se não conseguir mover-se passa ao movimento seguinte na sequência.** Note que cada *stormtrooper* individual de uma dada cor, consoante a sua localização no labirinto, pode estar em estados diferentes.

<i>Stormtrooper</i>	Sequência de movimentos			
B	cima	baixo	esquerda	direita
W	esquerda	direita	baixo	cima
O	direita	baixo	esquerda	cima

A aplicação deve então permitir:

- 1. Leitura do labirinto do jogo** (comando UPLOAD). São fornecidos dois inteiros **l** e **c**, com a altura e largura do labirinto. Seguem-se **l** linhas, cada uma delas com **c** caracteres, representando o conteúdo do labirinto. Em caso de sucesso é apresentada a mensagem (Maze accepted.).
A operação falha se: (1) já existir um labirinto definido (Maze already defined.).
- 2. Mostrar labirinto do jogo** (comando PRINT). Imprime o labirinto, no seu estado actual, incluindo informação sobre a localização dos personagens caso existam. Por exemplo, o labirinto deve apresentar carácter 'O' na posição (1,2) se existir nessa posição um *stormtroopers*. Em caso de sucesso, o labirinto é apresentado. Caso o jogo já tenha sido

iniciado, o labirinto é seguido de uma linha com a mensagem (Points: <N> Timer: <N> Rebels: <N> Game: <STATE>.), que mostra o número de pontos actuais, o tempo corrente, o número de rebeldes ainda activos e o estado do jogo (SETUP, ON ou OVER).

A operação falha se: (1) não existir um labirinto definido (Maze is undefined.).

3. **Adicionar soldado rebelde** (comando REBEL). São fornecidos o nome do rebelde e dois inteiros, a linha e a coluna, indicando a posição onde o personagem será criado. Em caso de sucesso esta operação adiciona o rebelde ao jogo (Rebel added.). O número de rebeldes está limitado pelo número de posições vazias do labirinto.

A operação falha se: (1) não existir um labirinto definido (Maze is undefined.); (2) o jogo já tiver sido iniciado (Game setup has already finished.); (3) já existir um rebelde com o nome dado (Rebel name already exists.); (4) a posição do labirinto estiver fora do labirinto ou não for uma posição vazia (Invalid maze position.).

4. **Iniciar jogo** (comando START). Este comando não recebe qualquer parâmetro. Em caso de sucesso o jogo é iniciado e mostra a mensagem (Points: <N> Timer: <N> Rebels: <N> Game: <STATE>.), que mostra o número de pontos actuais (0), o tempo corrente (0), o número de rebeldes activos, e o estado do jogo (ON).

A operação falha se: (1) não existir um labirinto definido (Maze is undefined.); (2) o jogo já tiver sido iniciado (Game setup has already finished.); (3) não existir nenhum soldado rebelde no jogo (Game needs a rebel.).

5. **Mover personagens** (comando MOVE). É fornecido o tipo de movimento (U, D, L, R) para cada um dos rebeldes ainda activos, segundo a ordem com que foram criados. Ou seja, têm de ser dados tantos movimentos quantos os rebeldes activos. Em caso de sucesso, esta operação movimenta todos os personagens, primeiro os rebeldes e depois os *stormtroopers*, ambos pela ordem que foram criados. Também incrementa o tempo por uma unidade e gera o output (Points: <N> Timer: <N> Rebels: <N> Game: <STATE>.), onde <STATE> representa o estado do jogo (ON ou OVER). Pode não ser possível mover um rebelde, porque a posição destino é uma parede ou está ocupada por outro soldado rebelde. O mesmo pode acontecer a um *stormtrooper*, se não se puder movimentar das quatro direcções. **Se o rebelde não se movimentar o número de passos não é alterado.**

A operação falha se: (1) não existir um jogo activo (Game is not on.).

São sempre fornecidos um número de movimentos que corresponde ao número de rebeldes activos.

6. **Listar personagem** (comando LIST <X>, onde <X> é ALL, REBELS, ou STORMTROOPERS). Em caso de sucesso a operação produz o cabeçalho (All avatars:), (Rebels:), ou (Stormtroopers:), respectivamente. O cabeçalho é seguido pela listagem dos vários personagens apresentando em cada linha o nome da personagem, a posição no labirinto, o seu estado (SUPERCHARGED, ACTIVE ou CAPTURED) e no caso dos rebeldes apresenta o número de pontos colectados. Na listagem ALL devem aparecer todos os personagens, mesmo os não activos. Enquanto nas restantes listagens (REBELS e STORMTROOPERS) devem aparecer apenas os personagens activos. Nas listagens os personagens devem ser apresentados por ordem de inserção. Note que os *stormtroopers* são inseridos antes dos rebeldes.

A operação falha se: (1) não existir um labirinto definido (Maze is undefined.); (2) não existirem personagens a listar (Nothing to list.).

7. **Listar caminho percorrido por soldado rebelde** (comando LIST PATH). É fornecido o nome do soldado. Como resultado a operação apresenta o caminho percorrido, da posição

mais antiga à mais recente. A listagem deve ser precedida por um cabeçalho (Rebel <NAME> has taken <N> steps and is <STATE>:), onde <NAME> é o nome do rebelde, <N> é o número de movimentos totais ~~(incluindo movimentos em que o rebelde permanece na mesma posição)~~ e <STATE> é ACTIVE, SUPERCHARGED, ou CAPTURED. De seguida as posições percorridas são apresentadas, uma por linha. Cada linha tem o formato (<row> <col>).

A operação falha se: (1) não existir um jogo activo ~~ou o jogo já tiver terminado~~ (Game has not started.); (2) não existir um rebelde com o nome dado (Rebel does not exist.).

8. **Listar rebeldes capturados por *stormtrooper*** (comando LIST PRISONERS). É fornecido o identificador do *stormtrooper*. Como resultado a operação apresenta, pela ordem de captura, o nome do rebelde e a localização onde foi capturado. A listagem deve ser precedida por um cabeçalho (Stormtrooper <ID> has captured <N> rebels and is <STATE>:), onde <ID> é o identificador do *stormtrooper*, <N> é o número rebeldes capturados e <STATE> é ACTIVE ou CAPTURED. De seguida os rebeldes capturados são apresentados, um por linha. Cada linha deve apresentar o nome do rebelde e a posição onde foi capturado.

A operação falha se: (1) não existir um jogo activo ~~ou o jogo já tiver terminado~~ (Game has not started.); (2) não existir um *stormtrooper* com o identificador dado (Stormtrooper does not exist.).

9. **Reiniciar o jogo** (comando RESET). Reinicia o jogo. Este comando tem sempre sucesso e pode ser executado em qualquer estado do jogo. Após a execução deste comando, terá que ser feito novamente o setup do jogo (leitura do labirinto e adição de rebeldes). ~~Em caso de sucesso o jogo é iniciado e mostra a mensagem (Game was reset.).~~

10. **Sair da aplicação** (comando EXIT). A operação tem sempre sucesso (Exiting.).

1.2. Exemplo de interacção com a aplicação

A aplicação desenvolvida tem que garantir o modelo de interacção ilustrado no exemplo seguinte (o carácter ↵ representa uma mudança de linha):

```
PRINT↵
Maze is undefined.↵
↵
UPLOAD↵
9 27↵
....OOO.....WWW.....↵
.....↵
P#####P.↵
..#.....#.....#.....↵
..###...#.....#.....↵
..#.....#.....#.....↵
..#.....#####.....#.....↵
.....↵
.....BBBB.....↵
Maze accepted.↵
↵
UPLOAD↵
2 4↵
#...↵
O...↵
Maze already defined.↵
```

```

↵
REBEL↵
Jyn↵
3 8↵
Rebel added.↵
↵
REBEL↵
Cassian↵
3 16↵
Rebel added.↵
↵
START↵
Points: 0 Timer: 0 Rebels: 2 Game: ON.↵
↵
MOVE↵
D D↵
Points: 20 Timer: 1 Rebels: 2 Game: ON.↵
↵
PRINT↵
..... O.....WWW .....↵
.....OO .....↵
P##### #####P.↵
..#....R.#....R....#....↵
..###....#.....#....↵
..#.....#.....#....↵
..#.....#####....↵
.....BBBB .....↵
.....↵
Points: 20 Timer: 1 Rebels: 2 Game: ON.↵
↵
LIST REBELS↵
Rebels:↵
Jyn 4 8 ACTIVE 10↵
Cassian 4 16 ACTIVE 10↵
↵
LIST PATH↵
Jyn↵
Rebel Jyn has taken 1 steps and is ACTIVE:↵
3 8↵
4 8↵
↵
EXIT↵
Exiting.↵
↵

```

2. Desenvolvimento

A sua aplicação deve tirar o melhor partido possível da matéria leccionada. Em particular, fazê-la **o mais extensível possível**. Deve ser construída de modo a **minimizar as alterações necessárias**, caso se pretendam acrescentar, mais tarde, **novos tipos de personagens** e labirintos compostos **por novos tipos de estruturas (portas, escadas, etc.)**.

Comece por desenvolver a interface principal da sua aplicação, identificando claramente quais os comandos que a sua aplicação deve suportar, bem como quais as entradas e saídas, não esquecendo as pré-condições. Depois, identifique as entidades de que vai necessitar para implementar este sistema. Identifique e especifique cuidadosamente as **interfaces** e **classes** de que necessita. Deve documentar a sua concepção e o seu desenvolvimento quer através de um diagrama de classes e interfaces, quer através de documentação adequada no seu código.

Construa o esqueleto da classe `Main` que trata da entrada e saída dos dados e da interacção com a aplicação. É normal que no princípio o seu programa ainda não faça tudo. Lembre-se da **regra da versão estável**: não tente fazer tudo de uma só vez. Vá fazendo, testando, e avançando por incrementos, à medida que as funcionalidades vão sendo desenvolvidas. Se necessário, crie pequenos programas de teste auxiliares. Desenvolva também as operações de forma incremental.

3. Submissão ao Mooshak

Para submeter o seu programa ao Mooshak registe-se no concurso **POO1617-TP1** e siga as instruções apresentadas na área Moodle da disciplina.

3.1. Sintaxe dos comandos

Note que para cada comando será apresentada apenas uma única mensagem de saída. Ou seja, as condições que causam falhas num comando devem ser verificadas pela ordem descrita no enunciado e assim que uma dessas condições se verificar não é necessário verificar as condições seguintes. Por exemplo, se ao executar o comando **REBEL** já existir um soldado rebelde com o nome dado, a mensagem de saída deve ser “Rebel name already exists.” e não é necessário verificar se a posição inicial para a personagem é válida.

3.2. Testes

Os testes do Mooshak estão divididos em 3 níveis. Na versão mínima há apenas um rebelde, sendo que não há poções nem *stormtroopers*. Na versão intermédia continua a haver apenas um rebelde, mas já há uma poção e um *stormtrooper* de cada cor. Na versão completa existem vários rebeldes, poções e *stormtroopers*.

3.2.1 Versão mínima (50 pontos)

- **Ficheiro de teste:** 1.1-upload-print_in.txt (10 pontos)
Comandos testados: UPLOAD, PRINT, EXIT
- **Ficheiro de teste:** 1.2-rebel_in.txt (10 pontos)
Comandos testados: REBEL
- **Ficheiro de teste:** 1.3-start_in.txt (10 pontos)
Comandos testados: START
- **Ficheiro de teste:** 1.4-move_in.txt (10 pontos)
Comandos testados: MOVE
- **Ficheiro de teste:** 1.5-list-rebels-path_in.txt (10 pontos)
Comandos testados: LIST REBELS, LIST PATH

3.2.2 Verão intermédia (30 pontos)

- **Ficheiro de teste:** 2.1-rebel-lists-captured_in.txt (10 pontos)
Comandos testados: UPLOAD, PRINT, REBEL, LIST REBELS, LIST STORMTROOPERS, LIST ALL, LIST PRISONERS
- **Ficheiro de teste:** 2.2-move-path_in.txt (10 pontos)
Comandos testados: LIST PATH
- **Ficheiro de teste:** 2.3-move-lists-captured_in.txt (10 pontos)
Comandos testados: testa a captura de *stormtroopers*

3.2.3 Verão completa (20 pontos)

- **Ficheiro de teste:** 3.1-game_in.txt (20 pontos)
Comandos testados: todos os comandos