



Índice

1	Regras.....	1
2	Descrição do Projeto.....	2
3	Relatório e Documentação.....	7
4	Critérios de Avaliação	8

1 Regras

- a) **O projeto é elaborado por grupos de 2 alunos** do mesmo docente de laboratório. Exceções têm de ser validadas pelo RUC.
- b) **A inscrição do grupo de projeto no Moodle é obrigatória.**
- c) **As discussões do projeto são obrigatórias.** As datas das discussões serão publicadas após a entrega final dos trabalhos.
- d) **Só serão considerados para avaliação projetos funcionais**, i.e., projetos que compilem sem erros sintáticos e que sejam executáveis. Projetos não compiláveis não serão avaliados.
- e) A apresentação de **relatórios ou implementações plagiadas leva à imediata atribuição de nota zero** a todos os trabalhos envolvidos, quer tenham sido o original ou as cópias.
- g) No relatório e nos ficheiros de implementação deverá constar o número, nome e turma de cada autor.
- h) Qualquer situação omissa neste enunciado e/ou nas regras de avaliação publicadas no SI é decidida pelo RUC.

1.1 Data e Forma de Entrega

O projeto é submetido no Moodle até às 10:00 de dia 2 de dezembro de 2024.

1. Projetos submetidos após esta data serão penalizados com 0,2 valores por cada hora de atraso até um máximo de 24 horas de atraso, após o que não serão considerados para avaliação.



O projeto é submetido no Moodle:

2. Através de um link de submissão - do respetivo docente de laboratório.
3. É submetido um único ficheiro, em formato .zip com o nome <numeroAluno1>_<numeroAluno2>.zip. (a utilização de outros formatos é penalizada)
4. O ficheiro em formato .zip deve conter o projeto IntelliJ, a documentação em JavaDoc e o relatório do projeto.

2 Descrição do Projeto

Pretende-se desenvolver um programa em Java para jogar o popular Minesweeper¹ (Minas & Armadilhas) em modo de consola.

No desenvolvimento do jogo deverá utilizar o paradigma de programação orientada por objetos, assim como os restantes conhecimentos de programação lecionados até ao momento nas aulas teórico-práticas e experimentados nas aulas prático-laboratoriais. **Não podem ser usadas relações entre classes, como composição ou herança, nem classes de coleção.**

2.1 Descrição da Lógica do Jogo

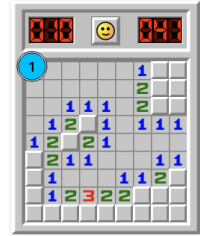
Tabuleiro e células

1. O jogo é jogado num tabuleiro bidimensional, composto por uma matriz 9x9 de células (Figura 1).
2. Cada célula do tabuleiro pode conter uma mina ou estar vazia.
3. O jogador deve descobrir células vazias e marcar as minas.

Regras do Jogo

1. O tabuleiro é inicializado com 10 minas, distribuídas aleatoriamente e ocultas do jogador – ver Figura 1.
2. O jogador escolhe células para abrir. Se a célula escolhida contiver uma mina, o jogo termina (o jogador perde) - ver Figura 2.
3. Se a célula escolhida não contiver uma mina, ela é aberta, revelando um número. Este número indica quantas minas estão nas oito células adjacentes – ver Figura 3.

¹ <https://minesweeper.online/>



4. Se a célula aberta não tiver minas adjacentes (o número é zero - pode ser omitido), todas as células adjacentes são automaticamente abertas, e essa operação continua “em cascata” até que apenas células adjacentes a minas permaneçam fechadas.
5. O jogador pode marcar células (ver Figura 3) que ele acredita conterem minas para evitar abri-las por engano – a marcação é feita com bandeiras, existindo tantas bandeiras quanto minas, i.e., 10.

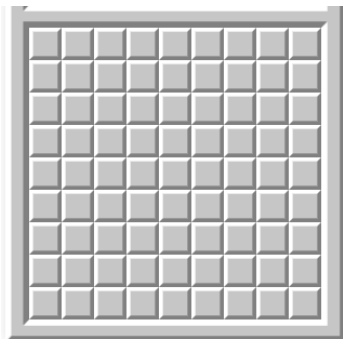


Figura 1 – Tabuleiro de jogo.

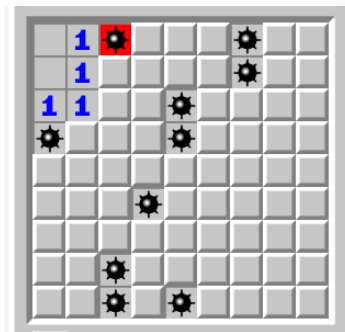


Figura 2 – Jogo perdido.

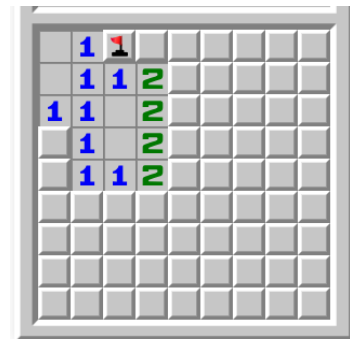


Figura 3 – Células abertas e mina marcada.

Objetivo do Jogo

1. O objetivo é abrir todas as células que não contêm minas.
2. O jogo é ganho quando todas as células seguras foram abertas e todas as minas foram corretamente marcadas.

2.2 Requisitos de Implementação

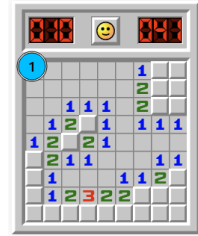
1. Ao iniciar a aplicação, deve existir um menu onde seja possível começar um jogo, visualizar os tempos das últimas 10 vitórias e terminar a aplicação, como se mostra na Figura 4:

```
MineSweeper Game
-----
1 - New Game
2 - Last 10 Wins
3 - Exit Game
Option>
```

Figura 4 – Exemplo de menu inicial



2. Antes do início de cada jogo deve ser pedido ao jogador a alcunha (nickname) com que vai jogar. Caso o jogador opte por não se registar com uma alcunha deve ser-lhe atribuída a alcunha “Anonymous #”. Onde # é um número sequencialmente atribuído.
3. No início de cada jogo deve ser apresentado ao jogador um tabuleiro 9x9 vazio com os comandos (ver Secção 2.2.1) passíveis de serem executados nesse contexto – ver Figura 5.
4. A cada célula corresponde uma coordenada do tabuleiro, onde as linhas são representadas por letras e as colunas por números. O jogador irá utilizar estas coordenadas para desencadear ações sobre as células.
5. Os caracteres utilizados para representar células por abrir, abertas, com bandeiras e com minas ficam ao critério do grupo de trabalho, mas estão limitados a caracteres ASCII.
6. Deve ser apresentado, junto do tabuleiro, a quantidade de bandeiras ainda disponíveis.
7. Deve ser apresentado, junto do tabuleiro, o tempo decorrido do jogo, em segundos.
8. Uma célula assinalada com uma bandeira não pode ser aberta diretamente (apenas pela abertura de casas adjacentes).
9. Quando um jogador perde, devem ser reveladas as posições de todas as minas. Deve ser perceptível no tabuleiro que mina explodiu, e.g., ver Figura 2.
10. Deve ser cronometrado o tempo que o jogador demora a ganhar um jogo e apresentá-lo no formato hh:mm:ss.
11. Quando um jogador vence o jogo, o seu tempo total deve ser apresentado na tabela das últimas 10 vitórias, no formato hh:mm:ss.
12. Os dados lidos da consola devem ser verificados e, caso se trate de um comando inválido ou com sintaxe inválida, o utilizador deve ser informado.



13. Toda a informação é mantida em memória durante a execução do programa, isto é, não se prevê nenhuma persistência de informação entre sessões (e.g., com ficheiros).
14. Situações omissas ficam ao critério do grupo de trabalho e devem ser mencionadas no relatório.

	0	1	2	3	4	5	6	7	8
A	□	□	□	□	1	■	■	■	■
B	□	□	□	□	1	2	■	■	■
C	□	1	1	1	□	1	1	■	■
D	□	1	#	1	□	□	1	■	■
E	1	2	2	1	□	□	1	■	■
F	1	#	1	1	1	1	1	■	■
G	1	1	1	1	■	■	■	■	■
H	□	□	1	3	■	■	■	■	■
I	□	□	1	■	■	■	■	■	■

Available flags: 8

Elapsed time: 9 seconds

[Type /help for assistance]

Command > _

Figura 5 - Tabuleiro de jogo após algumas jogadas. Representação em consola do jogo (esquerda); representação gráfica equivalente (direita).

2.2.1 Comandos

Os comandos são palavras reservadas antecidas pelo caracter '/'. Os comandos disponíveis, sua sintaxe e descrição correspondentes são apresentados na Tabela 1.



Tabela 1: Sintaxe e descrição dos comandos

Comando e Sintaxe	Descrição
/help	Apresenta a lista de comandos, a sua função e utilização.
/open <linha> <coluna>	Abre a célula nas coordenadas de tabuleiro linha/coluna, e.g., /open A 2.
/flag <linha> <coluna>	Marca a célula nas coordenadas de tabuleiro linha/coluna com uma bandeira. Se já existir uma bandeira nessa célula, remove-a.
/hint	Sugere ao jogador, de forma aleatória, uma célula que não contém minas. A indicação é feita na forma de coordenadas de tabuleiro.
/cheat	Comuta o jogo para modo de “batota”, onde as minas são reveladas a cada mostragem do tabuleiro.
/quit	Termina o jogo e volta para o menu principal. Um jogo assim terminado não entra na lista de vitórias. Deverá ser solicitada ao jogador a confirmação de que pretende terminar o jogo em curso.

💡 Pode fazer o *parse* do comando, e.g., "/open B 2" com o método `String.split()` em Java. Use o espaço como delimitador para separar as partes do comando num array de `String`. Exemplo:

```
String comando = "/open B 2";
```

```
String[] partes = comando.split(" ");
```

Neste caso, `partes[0]` será "/open", `partes[1]` será "B", e `partes[2]` será "2".



3 Relatório e Documentação

3.1 Documentação do Código

O código deve ser todo documentado utilizando JavaDoc e gerados os respetivos ficheiros com a documentação.

3.2 Relatório

Deverá produzir um pequeno relatório que contenha:

1. Capa

Título do trabalho, número(s) e nome(s) do(s) aluno(s), turma(s) e docente a quem se destina.

2. Introdução

Descrição breve do projeto e objetivo principal

3. Descrição da Implementação (1-2 páginas)

4. Estrutura Geral do Programa

Descrever como o código foi organizado (ex.: classes principais, métodos importantes).

5. Funcionalidades Implementadas

Listar as principais funcionalidades do jogo, como inicialização do tabuleiro, abertura de células, e deteção de minas.

6. Decisões de Design

Explicar qualquer escolha significativa feita durante a implementação, como simplificações na lógica de jogo.

7. Exemplo de Execução

Descrever um exemplo simples de como o jogo funciona, com capturas de ecrã ou uma descrição da interação.

8. Conclusão

Reflexão sobre o desenvolvimento do projeto e o que foi aprendido.

9. Referências (se necessário)

Citar qualquer recurso utilizado, como tutoriais ou documentação oficial de Java.



4 Critérios de Avaliação

Será aplicada a grelha de avaliação em 4.1 e a nota final será ponderada pela qualidade da discussão (considerando os conhecimentos sobre programação e sobre o projeto demonstrados por cada aluno/a). Existirá um pequeno questionário inicial e individual na discussão. Consequentemente, a nota final poderá ser diferente entre os elementos do grupo.

4.1 Grelha de Avaliação

O projeto é avaliado de acordo com a grelha de avaliação apresentada na Tabela 2 e penalizações descritas na Tabela 3.

Os valores da Tabela 2 indicam a pontuação máxima de cada item, a pontuação atribuída terá em consideração a qualidade da implementação e do código escrito. A nota final será obtida pela ponderação dos elementos de avaliação.

Tabela 2: Grelha de avaliação

Item e Ponderação	Cotação Máxima (valores)
Documentação JavaDoc (10%)	20
Relatório (10%)	20
Implementação (70%)	20
Tabuleiro (representação interna e colocação de minas)	2
Tabuleiro (apresentação na consola)	1
Validação de <i>inputs</i> do utilizador	1
Registo de jogadores	1
Histórico de vitórias	2
Navegação entre menus de jogo (e.g., New Game e /quit)	1
Comando /help	0.5
Comando /hint	0.5
Comando /cheat	1
Comando /flag	2
Algoritmo do motor de jogo, inclui comando /open	8
Questionário que antecede discussão (10%)	20



4.2 Grelha de Penalizações

Aplicam-se as penalizações descritas na Tabela 3.

Tabela 3: Principais Penalizações

Item	Penalização
Relatório plagiado (1)	Anulação
Projeto plagiado (1)	Anulação
Não comparência à discussão	0 (p/ aluno)
Projeto com erros sintáticos; não compilável	0 (p/ projeto)
Submissão por grupo não esteja inscrito no Moodle	-3.0 val
Projeto que termine abruptamente (2)	-1 val
Indentificador que não respeite convenções do Java (p/ cada um)	-0.1 val
Indentação inconsistente (p/ ficheiro)	-0.5 val
Código não documentado (p/ função/método)	-0.5 val
Nome de ficheiro errado (3)	-0.5val
Formato de ficheiro submetido compactado com formato errado (4)	-1 val

(1) Todos, independentemente de serem o original ou o copiado.

(2) E.g., erros de “IndexOutOfBoundsException” ou “NullPointerException”. Exceptuam-se “InputMismatchException”.

(3) Que não respeito o formato <numeroDeAluno1>_<numeroDeAluno2>.zip

(4) Que esteja compactado em qualquer outro formato que não o .zip.