

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Ciência da Computação

Campo Minado

BCC221 - Programação Orientada a Objetos

Tiago Mol, Vinicus Anjos, Luiz Eduardo, Leandro Augusto
Professor: Guillermo Cámara-Chávez

Ouro Preto
4 de fevereiro de 2024

Sumário

1	Introdução	1
1.1	Objetivos Específicos	1
1.2	Justificativa	1
1.3	Ferramentas utilizadas	1
1.4	Especificações da máquina	1
2	Desenvolvimento	2
2.1	Programa Principal e Construção da Tela	2
2.2	Botões	2
2.3	Funcionalidades Adicionais	2
2.4	Especificações de Requisitos	2
2.5	Diagrama das Classes	3
3	Conclusão	4
4	Link do Video	5

Lista de Figuras

1	Diagrama	3
---	--------------------	---

1 Introdução

Utilizando a linguagem de programação Java e a biblioteca gráfica Swing, buscamos desenvolver uma aplicação interativa que não apenas recrie a experiência tradicional do jogo, mas também acrescente elementos visuais e funcionais que enriqueçam a experiência do usuário.

1.1 Objetivos Específicos

- Configuração do Campo Minado: Minas posicionadas aleatoriamente em um campo 9x9 com 10 minas.
- Botão esquerdo revela, direito marca/desmarca minas.
- Contador de minas
- Botão de reinício com "Cara feliz/triste" indica o estado do jogo.
- Final do jogo revela todo o campo e exibe caixa de diálogo.
- Possibilidade de jogar com mouse e teclado.
- Modulo Pânico
- Utilização de componentes Swing

1.2 Justificativa

Este trabalho foi proposto e desenvolvido visando a aplicação e cristalização do conhecimento absorvido durante as aulas de Programação Orientada a Objetos (BCC221). Como uma simulação de uma aplicação no mundo real e usual, este trabalho possibilitou aprofundar no mundo no qual a ementa da disciplina nos apresenta.

1.3 Ferramentas utilizadas

- Ambiente de desenvolvimento do código fonte: VScode.
- Linguagem utilizada: Java.
- Ambiente de desenvolvimento da documentação: Overleaf \LaTeX .¹

1.4 Especificações da máquina

A máquina onde o desenvolvimento e os testes foram realizados possui a seguinte configuração:

- Processador: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz 2.90 GHz.
- Memória RAM: 16Gb.
- Sistema Operacional: Linux.

Para a compilação do projeto, basta digitar:

Compilando o projeto

```
javac Minesweeper.java java Minesweeper
```

¹Disponível em <https://www.overleaf.com/>

2 Desenvolvimento

O código do jogo Campo Minado em Java utiliza a biblioteca Swing para criar uma interface gráfica interativa através das principais funcionalidades oferecidas pela interface, com o uso de telas "JFrame" para a construção do tabuleiro, e botões "JButton" para as funcionalidades do jogo em si, além da lógica implementada por trás de cada artifício. O jogo segue os padrões clássicos do Campo Minado, tais como exemplificadas na documentação do trabalho. O código está estruturado em classes e métodos, incorporando os conceitos de orientação a objetos e interação com eventos de interface. Quanto ao seu desenvolvimento, foi guiado por uma abordagem iterativa, enfatizando a modularidade e testes abrangentes, de forma a garantir a correção e a usabilidade do software. O jogo é iniciado com a criação de um tabuleiro de nível simples, onde dez minas são distribuídas aleatoriamente. A interação do jogador é implementada com cliques do mouse e, para alguns botões, também do teclado. Tais cliques permitem revelar as células ou marcar os possíveis locais de minas.

2.1 Programa Principal e Construção da Tela

Para a tela do jogo, foi utilizada a ferramenta "JFrame" através do módulo principal do jogo, chamado "Minesweeper.java". O módulo, por sua vez, é uma extensão da classe JFrame, que herda seus atributos e implementa as funcionalidades e atributos adicionais necessários para que o jogo opere como desejado, tais como vetores de células, métodos ActionListener e até mesmo a implementação das definições da célula em si.

2.2 Botões

Para a implementação dos botões do jogo, foi utilizada a ferramenta "JButton", fornecida pela própria interface gráfica Swing. A ferramenta em questão corresponde a botões de dois estados, ativos ou não, com funcionalidades implementadas por trás de cada um destes estados. Para os botões correspondentes às áreas jogáveis, estes podem estar revelados ou não, e quando revelados, devem indicar a quantidade de minas vizinhas. Também foram utilizados "JButton"s para o jogador reiniciar o jogo, desistir do jogo atual ou encerrar o programa.

2.3 Funcionalidades Adicionais

Algumas funcionalidades adicionais e alterações do jogo foram implementadas, visando atender as demandas da documentação do trabalho. Entre estas, estão presentes o "Módulo Pânico": um botão que preenche a tela com uma imagem qualquer quando o mouse adentra o seu território, e a exclusão do contador de tempo desde o início do jogo, que é uma funcionalidade significativa presente no jogo original. Além disso, o jogo possui funcionalidades de contagem de minas, atualização dinâmica do contador na interface e checagem de vitória ou derrota.

2.4 Especificações de Requisitos

O jogo em questão é relativamente simples, e dada a sua arquitetura bidimensional, implementada inteiramente em java, o jogo não tem requisitos de hardware mínimos. Para a execução do jogo, entretanto, é necessário que o usuário instale os pacotes necessários para a compilação do código Java.

2.5 Diagrama das Classes

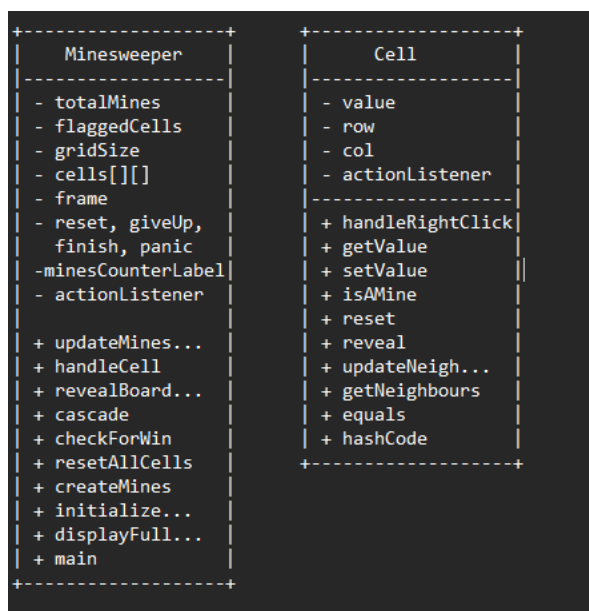


Figura 1: Diagrama

Este é um diagrama simplificado, e você pode notar que o código possui duas classes principais:

- Minesweeper: A classe principal que estende JFrame e representa o jogo Minesweeper. Ela contém os botões, a grade de células e métodos para lidar com eventos e lógica do jogo.
- Cell: Uma classe interna que estende JButton e representa uma célula no jogo. Cada célula tem sua posição (linha e coluna), um valor associado (indicando se é uma mina ou o número de minas vizinhas), métodos para manipular o clique do botão e métodos para atualizar o estado da célula.
- Além disso, há métodos como `redefineAndInsertImage`, `updateMinesCounterLabel`, `handleRightClick`, `initializeButtonPanel`, `initializeGrid`, `handleGridRightClick`, entre outros, que são utilizados para realizar diferentes funcionalidades no jogo. Esses métodos estão na classe principal (Minesweeper) e ajudam a organizar a lógica do jogo.

3 Conclusão

Na conclusão, destaco que a implementação do jogo Campo Minado reflete uma abordagem eficaz e estruturada para a criação de uma versão simplificada e funcional do clássico jogo. A utilização de classes e métodos bem definidos permite uma organização clara da lógica do jogo, facilitando a compreensão e manutenção do código.

A introdução de uma classe principal, Minesweeper, encapsula a lógica do jogo e promove a modularidade, tornando mais fácil estender ou modificar o sistema no futuro. O uso de classes internas, como a classe Cell, contribui para a encapsulação e reutilização de código, seguindo boas práticas de orientação a objetos.

A implementação utiliza recursos gráficos da biblioteca Swing para criar uma interface simples e interativa, oferecendo uma experiência de usuário amigável. Além disso, a introdução de recursos como o botão de pânico e a exibição de imagens em tela cheia adicionam elementos de diversão e originalidade ao jogo.

A lógica do jogo, incluindo a manipulação de cliques, a revelação das células e a verificação de vitória, está bem estruturada. A abordagem polimórfica utilizada, especialmente na classe Cell, permite tratar diferentes tipos de células de maneira uniforme, contribuindo para a flexibilidade do código.

Em resumo, o desenvolvimento desta versão simplificada do Campo Minado atende não apenas aos requisitos funcionais do jogo, mas também incorpora boas práticas de programação, modularidade e interatividade. A implementação bem-sucedida reflete um entendimento sólido dos conceitos de programação orientada a objetos e da manipulação de interfaces gráficas em Java.

4 Link do Video

Para acessar, basta entrar com email vinculado a UFOP.

<https://drive.google.com/file/d/1m0F3kII2Lu1lw1EBwh2uJ3BEP31-1oW0/view?usp=sharing>