

Présentation

Projet jeu démineur en java

Clement Thavarajan

SOMMAIRE

Présentation du code

- Préparation
- Compilation du code
- Exécution du programme

Importation des packages et déclaration de la classe

Case et de la classe Grille

- Dans cette partie, nous importons les packages nécessaires et déclarons une classe interne Case. La classe Case représente chaque case de la grille du jeu de
- démineur. Elle contient trois attributs : mine pour indiquer si la case contient une
- mine, discovered pour indiquer si la case a été découverte, et adjacentMines pour stocker le nombre de mines adjacentes à cette case. Nous déclarons la classe principale Grille, qui hérite de la classe JFrame de Swing. Cette classe représente
- la grille du jeu de démineur. Elle contient des attributs pour la taille de la grille, le nombre de mines, la grille de cases et la grille de boutons.

-
-

```
package Vue;

import javax.swing.*;
import java.awt.*;
import java.util.Random;

class Case {
    boolean mine;
    boolean discovered;
    int adjacentMines;

    public Case() {
        this.mine = false;
        this.discovered = false;
        this.adjacentMines = 0;
    }
}

public class Grille extends JFrame {
    Case[][] grille;
    int size;
    int mines;
    JButton[][] boutons;

    public Grille (int size, int mines) {
        this.size = size;
        this.mines = mines;
        this.grille = new Case[size][size];
        this.boutons = new JButton[size][size];

        setLayout(new GridLayout(size, size));
    }
}
```

Initialisation de la grille et des boutons

Dans cette partie, nous initialisons la grille et les boutons du jeu. Nous parcourons chaque case de la grille et créons un bouton correspondant.

Chaque bouton est configuré pour exécuter une action lorsqu'il est cliqué. L'action consiste à afficher le contenu de la case correspondante et à gérer les interactions du joueur.

Fonction de placement des mines

Cette fonction place aléatoirement un certain nombre de mines sur la grille.

Elle utilise la classe Random pour générer des coordonnées aléatoires et placer les mines dans des cases non minées.

Calcul du nombre de mines adjacentes

Cette fonction calcule le nombre de mines adjacentes à chaque case de la grille.

Elle parcourt chaque case et examine ses huit voisins pour compter le nombre de mines.

```

private void resetGrille() {
    new Grille(6, 6);
}

private void placeMines() {
    Random rand = new Random();
    int count = 0;
    while (count < mines) {
        int x = rand.nextInt(size);
        int y = rand.nextInt(size);
        if (!grille[x][y].mine) {
            grille[x][y].mine = true;
            count++;
        }
    }
}

private void calculateAdjacentMines() {
    int[] dx = {-1, -1, -1, 0, 0, 1, 1, 1};
    int[] dy = {-1, 0, 1, -1, 1, -1, 0, 1};

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (!grille[i][j].mine) {
                for (int k = 0; k < 8; k++) {
                    int x = i + dx[k];
                    int y = j + dy[k];
                    if (isValid(x, y) && grille[x][y].mine) {
                        grille[i][j].adjacentMines++;
                    }
                }
            }
        }
    }
}
}
}

```

```

private boolean isValid(int x, int y) {
    return x >= 0 && x < size && y >= 0 && y < size;
}

public static void main(String[] args) {
    new Grille(6, 6);
}

```

Fonction pour vérifier la validité des coordonnées

Cette fonction vérifie si les coordonnées spécifiées sont valides dans la grille.

Elle retourne vrai si les coordonnées sont dans les limites de la grille et faux sinon.

Fonction main

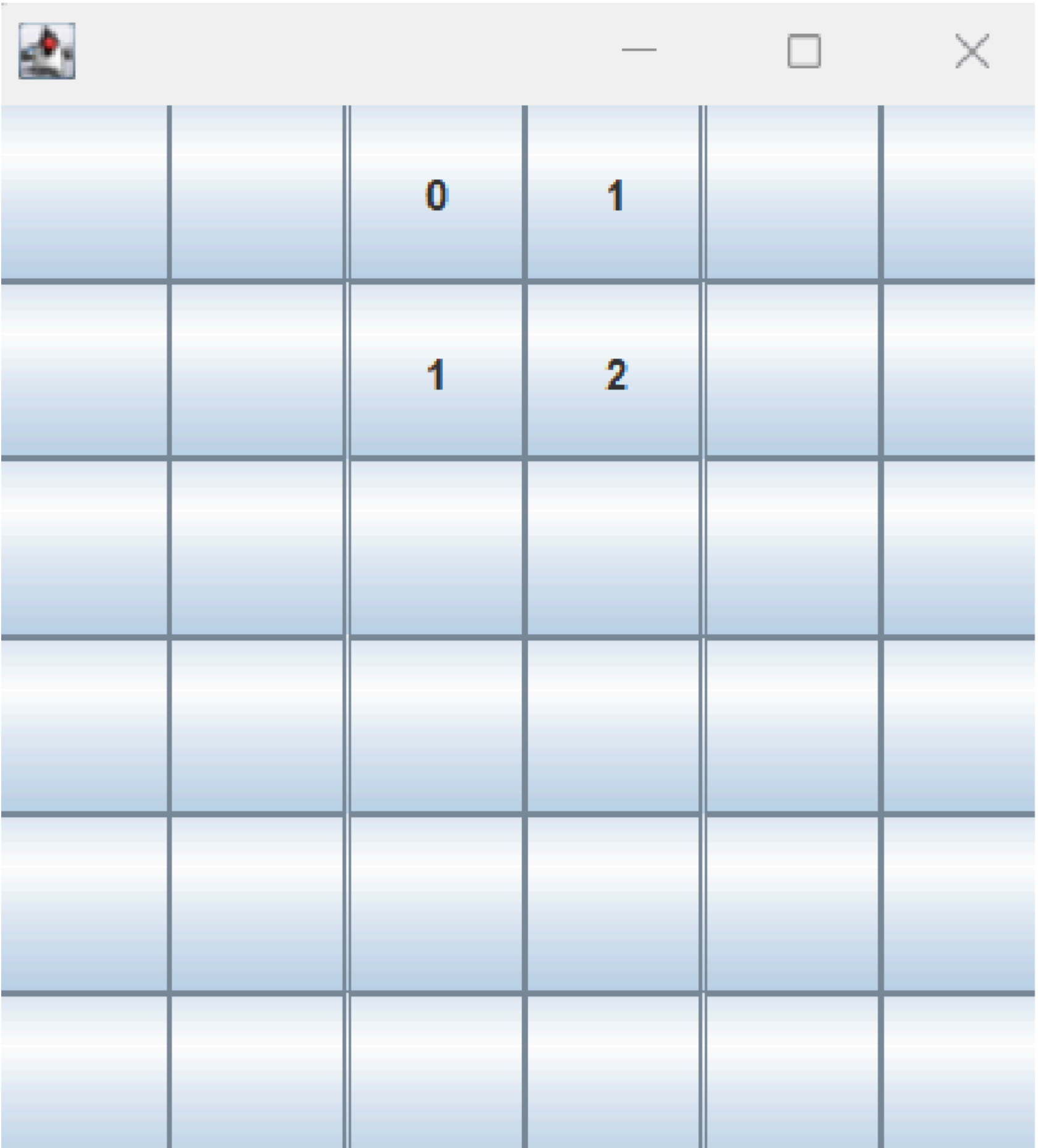
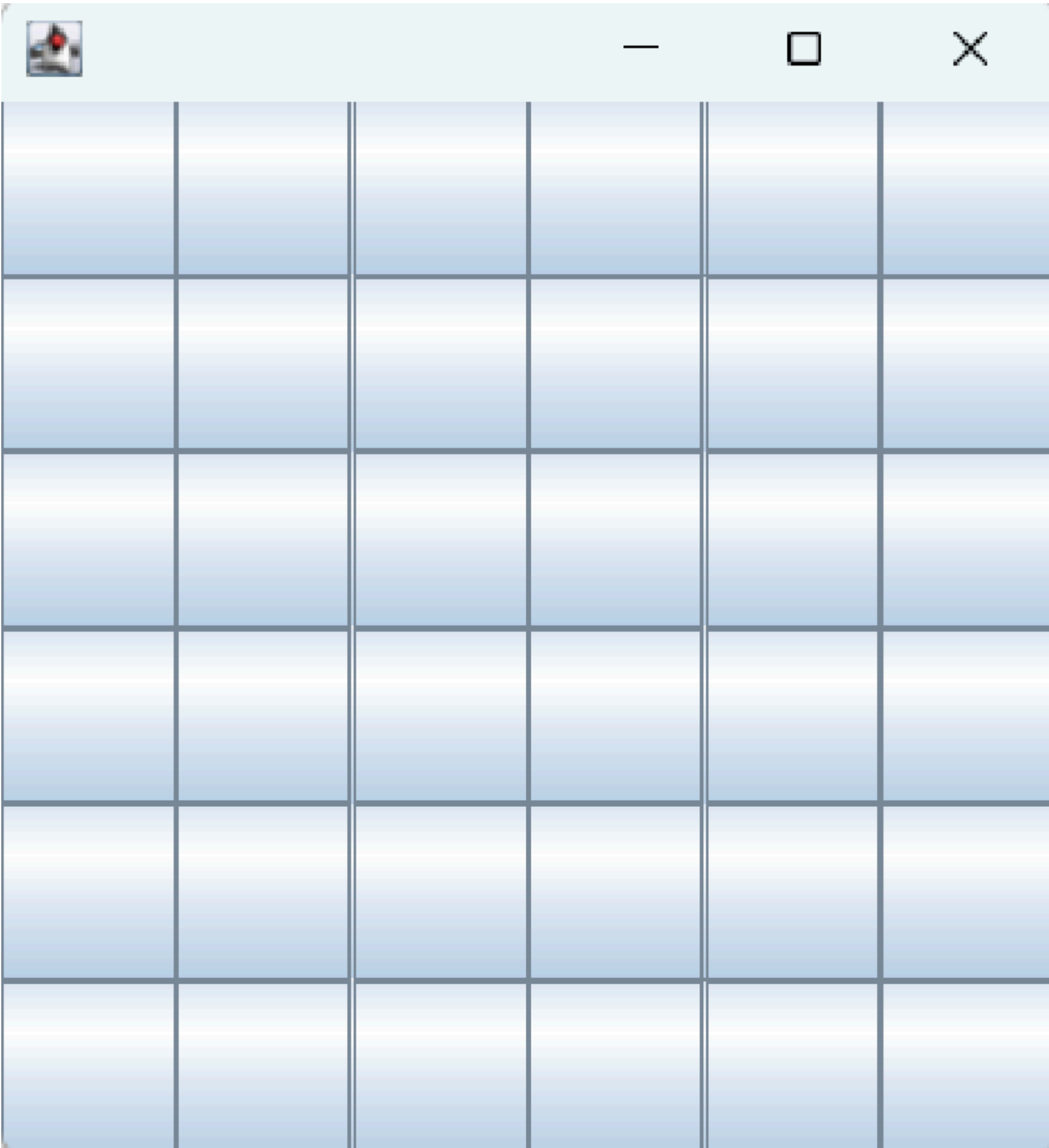
Fonction main

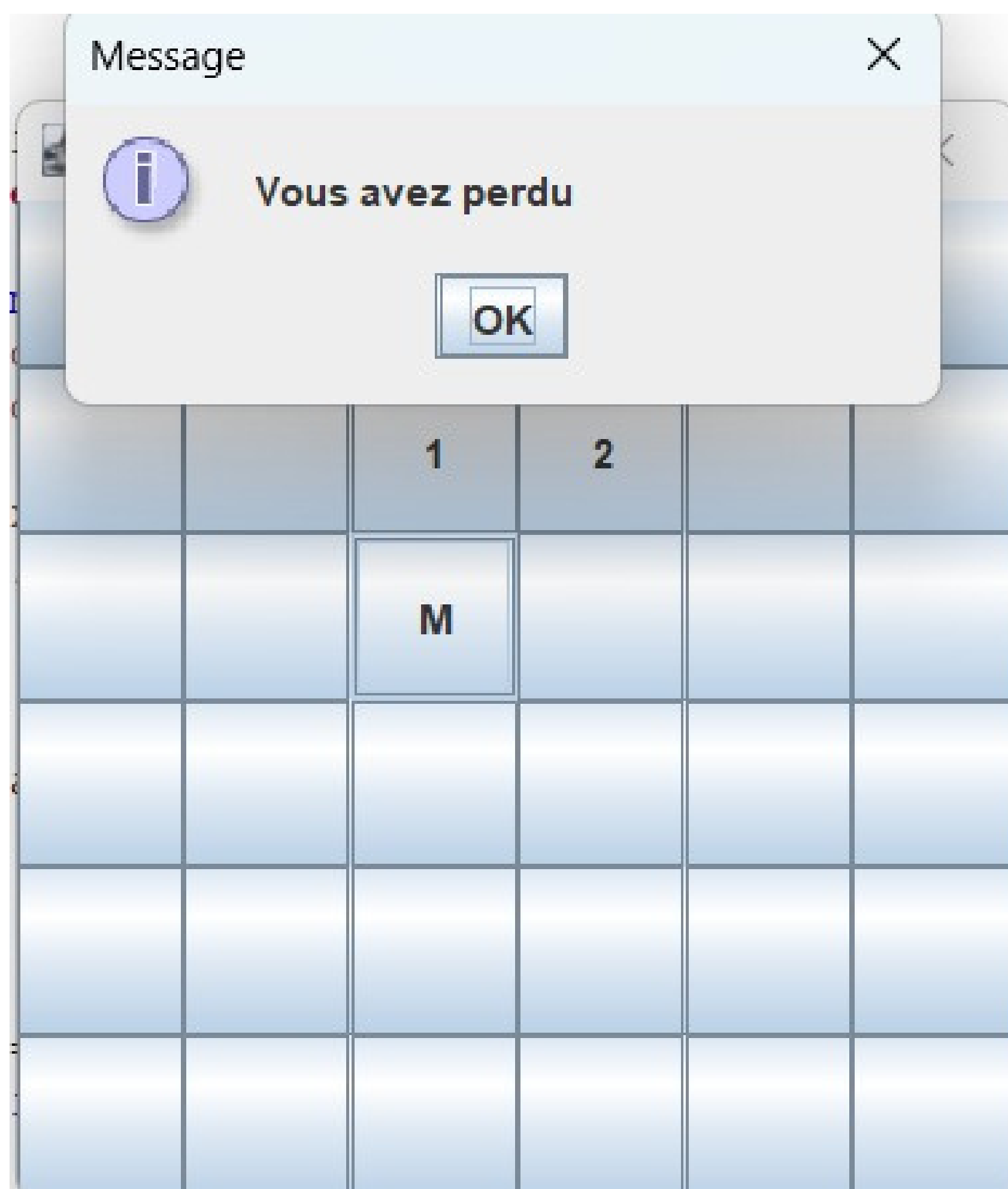
La fonction main crée une instance de la classe Grille avec une taille de grille et un nombre de mines spécifiés, ce qui lance le jeu de démineur.

```
for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) {
        grille[i][j] = new Case();
        boutons[i][j] = new JButton();
        boutons[i][j].setPreferredSize(new Dimension(50, 50));
        boutons[i][j].addActionListener(e -> {
            // Mettre ici l'action à effectuer lorsqu'un bouton est cliqué
            JButton button = (JButton) e.getSource();
            int x = -1, y = -1;
            for (int k = 0; k < size; k++) {
                for (int l = 0; l < size; l++) {
                    if (button.equals(boutons[k][l])) {
                        x = k;
                        y = l;
                        break;
                    }
                }
            }
            if (x != -1) break;
        });
        // Afficher le contenu de la case
        if (grille[x][y].mine) {
            button.setText("M");
            JOptionPane.showMessageDialog(this, "Vous avez perdu ");
            resetGrille();
        } else {
            button.setText(Integer.toString(grille[x][y].adjacentMines));
        }
    }
}
pack();
setVisible(true);

placeMines();
calculateAdjacentMines();
}
```

Aperçu





Merci pour votre attention

