Sure! Below is a simple implementation of Minesweeper using Java Swing. This implementation includes a basic GUI for the game. Note that this is a basic version, and you can extend it further to include features like timing, scoring, difficulty levels, etc.

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Minesweeper extends JFrame {
    private final int ROWS = 10;
    private final int COLS = 10;
    private final int NUM_MINES = 10;

    private JButton[][] buttons;
    private boolean[][] isMine;
    private boolean[][] isRevealed;
    private int[][] neighborMines;

    public Minesweeper() {
        setTitle("Minesweeper");
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(ROWS, COLS));

        buttons = new JButton[ROWS][COLS];
        isMine = new boolean[ROWS][COLS];
        isRevealed = new boolean[ROWS][COLS];
        neighborMines = new int[ROWS][COLS];

        initializeGame();

        setVisible(true);
    }

    private void initializeGame() {
        // Place mines randomly
        for (int i = 0; i < NUM_MINES; i++) {
            int row, col;
            do {
                row = (int) (Math.random() * ROWS);
                col = (int) (Math.random() * COLS);
            } while (isMine[row][col]);
            isMine[row][col] = true;
        }

        // Initialize buttons and their action listeners
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                JButton button = new JButton();
                button.setPreferredSize(new Dimension(30, 30));
                button.setFocusPainted(false);
                button.addMouseListener(new ButtonListener(i, j));
                add(button);
                buttons[i][j] = button;
            }
        }
    }

    private int countNeighborMines(int row, int col) {
        int count = 0;
        for (int i = Math.max(0, row - 1); i <= Math.min(row + 1, ROWS - 1); i++) {
            for (int j = Math.max(0, col - 1); j <= Math.min(col + 1, COLS - 1); j++) {
                if (isMine[i][j]) {
                    count++;
                }
            }
        }
        return count;
    }

    private void revealNeighbors(int row, int col) {
        for (int i = Math.max(0, row - 1); i <= Math.min(row + 1, ROWS - 1); i++) {
            for (int j = Math.max(0, col - 1); j <= Math.min(col + 1, COLS - 1); j++) {
                if (!isRevealed[i][j] && !isMine[i][j]) {
                    revealCell(i, j);
                }
            }
        }
    }

    private void revealCell(int row, int col) {
        isRevealed[row][col] = true;
        buttons[row][col].setEnabled(false);
        buttons[row][col].setBackground(Color.LIGHT_GRAY);
        if (neighborMines[row][col] > 0) {
            buttons[row][col].setText(Integer.toString(neighborMines[row][col]));
        } else {
            revealNeighbors(row, col);
        }
    }

    private class ButtonListener extends MouseAdapter {
        private int row, col;

        public ButtonListener(int row, int col) {
            this.row = row;
            this.col = col;
        }

        @Override
        public void mouseClicked(MouseEvent e) {
            if (SwingUtilities.isRightMouseButton(e)) {
                // Right click to mark/unmark mine
                JButton button = (JButton) e.getComponent();
                if (!isRevealed[row][col]) {
                    if (button.getText().equals("")) {
                        button.setText("X");
                    } else {
                        button.setText("");
                    }
                }
            } else {
                // Left click to reveal cell
                if (!isRevealed[row][col]) {
                    if (isMine[row][col]) {
                        JOptionPane.showMessageDialog(null, "Game Over! You hit a mine!");
                        revealAllMines();
                    } else {
                        revealCell(row, col);
                    }
                }
            }
        }
    }

    private void revealAllMines() {
        for (int i = 0; i < ROWS; i++) {
            for (int j = 0; j < COLS; j++) {
                if (isMine[i][j]) {
                    buttons[i][j].setEnabled(false);
                    buttons[i][j].setBackground(Color.RED);
                }
            }
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new Minesweeper());
    }
}
```

This code sets up a basic Minesweeper game with a 10x10 grid and 10 mines. You can click on the cells to reveal them and right-click to mark a cell as potentially containing a mine. When you hit a mine, a message dialog will appear, and all mines will be revealed.