# **Minesweeper Game - Documentation**

## **Overview**

## This is a console-based Minesweeper game implemented in Java. It follows object-oriented principles and is designed with separation of concerns, adhering to the Single Responsibility Principle (SRP). The game randomly places mines on a grid and challenges the user to uncover cells without hitting a mine. The game provides feedback at each step and allows for replay after a game is won or lost.

## 

## **Design and Assumptions**

### **Design**

1. **Classes**:
   * **MinesweeperGame**: The main class orchestrating the game flow, managing user inputs, and integrating other components.
   * **UserInterface**: Handles all interactions with the user, including displaying the grid, capturing input, and showing messages.
   * **GameGrid**: Manages the grid, placing mines, calculating adjacent mine counts, and processing cell reveals.
   * **Utility**: Contains reusable utility functions such as input validation and constants.
   * **GameException**: Custom exception class for handling game-specific errors.
2. **Key Features**:
   * Randomized mine placement for variability in gameplay.
   * Grid display updates dynamically based on user inputs.
   * Automatic uncovering of adjacent cells if no mines are nearby.
   * Game states:
     + Game Over: When a mine is uncovered.
     + Game Won: When all non-mine cells are uncovered.
   * Replay prompt to continue playing after a game ends.
3. **Assumptions**:
   * Valid user inputs are assumed (handled via validation methods).
   * The game is played on square grids of size 2x2 or larger.
   * The maximum number of mines is capped at 35% of the grid size for a balanced experience.

## **Design and Assumptions**

### **Design**

1. **Classes**:
   * **MinesweeperGame**: The main class orchestrating the game flow, managing user inputs, and integrating other components.
   * **UserInterface**: Handles all interactions with the user, including displaying the grid, capturing input, and showing messages.
   * **GameGrid**: Manages the grid, placing mines, calculating adjacent mine counts, and processing cell reveals.
   * **Utility**: Contains reusable utility functions such as input validation and constants.
   * **GameException**: Custom exception class for handling game-specific errors.
2. **Key Features**:
   * Randomized mine placement for variability in gameplay.
   * Grid display updates dynamically based on user inputs.
   * Automatic uncovering of adjacent cells if no mines are nearby.
   * Game states:
     + Game Over: When a mine is uncovered.
     + Game Won: When all non-mine cells are uncovered.
   * Replay prompt to continue playing after a game ends.
3. **Assumptions**:
   * Valid user inputs are assumed (handled via validation methods).
   * The game is played on square grids of size 2x2 or larger.
   * The maximum number of mines is capped at 35% of the grid size for a balanced experience.

## **Environment Setup**

### **Prerequisites**

* **Java Version**: Java 17 or later
* **Dependency Management** : Maven
* **Operating System**: Windows, macOS, or Linux
* **Development Tools**:
  + Command line/terminal
  + Any text editor or IDE (e.g., IntelliJ IDEA, Eclipse, or VS Code)

## **How to Run the Application**

### **Steps**

1. **Extract folder and go in location**
2. **Compile the Code**:
   * Open a terminal or command prompt.  
     Run following commands  
     **mvn clean install ( will run tests as well)  
     cd target**
   * **java -jar MineSweeper.jar**
3. Application will start running

**Play**:

* Follow on-screen instructions:
  + Enter grid size.
  + Specify the number of mines.
  + Input cell coordinates to reveal.
  + Type q to quit at any time.

## **Functionalities**

### **1. Starting the Game**

* Displays a welcome message and prompts the user to input grid size and number of mines.

### **2. Grid Display**

* Shows the current state of the grid with:
  + \_ for unrevealed cells.
  + Numbers for revealed cells indicating the count of adjacent mines.
  + \* for mines (revealed only at the end of the game).

### **3. Gameplay**

* **Revealing Cells**:
  + Input the coordinates of the cell (e.g., A1).
  + Revealed cells may trigger:
    - Chain reveals if no adjacent mines exist.
    - End game if a mine is revealed.
* **Game End States**:
  + *Game Over*: When a mine is uncovered.
  + *Game Won*: When all non-mine cells are revealed.

### **4. Replay**

* Prompts the user to play again or quit.

## **Example Interaction**

text

Copy code

Welcome to Minesweeper!

Enter grid size (e.g., 5 for a 5x5 grid): 5

Enter the total number of mines (maximum 8): 3

1 2 3 4 5

A \_ \_ \_ \_ \_

B \_ \_ \_ \_ \_

C \_ \_ \_ \_ \_

D \_ \_ \_ \_ \_

E \_ \_ \_ \_ \_

Select a cell to reveal (e.g., A1): A1

1 2 3 4 5

A 1 \_ \_ \_ \_

B \_ \_ \_ \_ \_

C \_ \_ \_ \_ \_

D \_ \_ \_ \_ \_

E \_ \_ \_ \_ \_

Select a cell to reveal (e.g., A1): B2

Boom! You hit a mine. Game Over.

Would you like to play again? (Press 'q' to quit or Enter to continue): q