## **Instructions**

### **Minesweeper Game Using a 2D Array**

#### **1. Game Overview:**

Minesweeper is a classic puzzle game where the player uncovers cells on a grid, trying to avoid hidden mines. The goal is to reveal all non-mine cells while marking all mines with flags.

#### **2. Game Board Representation:**

* The game board is represented by a 2D array where each cell can either be empty, contain a number (indicating the number of adjacent mines), or contain a mine.
* For example, a 10x10 grid would be represented by a board[10][10] array.

#### **3. Initialization:**

* The board is initialized with all cells hidden.
* Mines are randomly placed in some of the cells. The number of mines depends on the difficulty level.
* After placing the mines, each non-mine cell is populated with a number that indicates how many mines are adjacent to it (including diagonals).

#### **4. Player Input:**

* Players can either reveal a cell or place a flag on a cell.
* If a player reveals a cell containing a mine, the game is over.
* If a player reveals a cell with no adjacent mines (number 0), the game will automatically reveal all adjacent cells recursively until it hits cells that are next to mines.

#### **5. Game Logic:**

* The game continuously checks if the player has revealed all non-mine cells, which means they’ve won.
* The game also checks if a revealed cell contains a mine, which results in a loss.

#### **6. Endgame Scenarios:**

* The player wins by successfully revealing all non-mine cells and correctly flagging all mines.
* The player loses if they reveal a cell containing a mine.

### **Conclusion:**

Minesweeper effectively utilizes a 2D array to manage the game board, track the location of mines, and calculate the number of adjacent mines for each cell. This project challenges developers to handle randomization, recursion, and efficient array management, making it an excellent example of using a 2D array in a game setting.