**Minesweeper Game – Milestone 1**

A console-based C# application simulating the foundational structure of a Minesweeper game. Users can view generated game boards with bomb placement and adjacent bomb counts. This milestone focuses on building class structure, implementing core logic, and visually displaying the internal board in the console.

**🎯 Milestone Goal**

The goal of this milestone is to implement the three core classes:

* Cell: Defines the structure and properties of each square on the board
* Board: Manages the grid, bomb placement, and calculation of neighbor bombs
* Program: Acts as the entry point, showcasing how the board and logic work

This foundation will support gameplay logic in future milestones.

**🔑 Key Features**

* 🎲 Randomized board generation with adjustable size and difficulty
* 💣 Bomb placement based on a percentage (difficulty value)
* 🧠 Calculation of number of bombs adjacent to each cell
* 🖥️ Color-coded console output to display bomb (B) and neighbor counts
* 🧱 Structured using object-oriented principles for clarity and extensibility

**🗂️ Project Structure**

MineSweeperClasses/

├── Cell.cs → Defines properties of each board cell

├── Board.cs → Handles grid creation, bomb setup, and neighbor counting

├── Program.cs → Main entry point for running the game

├── UML Diagram.png → Visual representation of class relationships

└── .gitignore → Excludes .vs/, bin/, obj/ from Git

Each component is separated by responsibility, following best practices for class-based programming.

**🚀 Getting Started**

**Requirements**

* .NET Core / .NET Framework
* Visual Studio or VS Code (with C# support)

**Steps to Run**

1. Clone or download the project files
2. Open the solution in Visual Studio or VS Code
3. Build the project to ensure there are no compile-time errors
4. Run the application (Program.cs)
5. View console output displaying two generated Minesweeper boards

**🧠 Summary**

This milestone showcases the object-oriented implementation of Minesweeper using C#. We structured the logic into three main classes. The Board class initializes a 2D array of Cell objects, places bombs randomly, and calculates bomb neighbor counts. The Cell class tracks the bomb status, visit state, and number of bomb neighbors. The Program class orchestrates board generation and visualization.

Key C# concepts demonstrated include:

* **Class encapsulation**
* **Constructors and property accessors**
* **2D array traversal**
* **Random number generation**
* **Conditional logic for board state calculations**

This sets the groundwork for future enhancements like cell reveal, flagging, and user interaction.

* Board bounds checked when counting neighboring bombs
* Uses Random class securely and efficiently