**🧩 Minesweeper Game – Milestone 2**

A **console-based interactive C# application** that builds on the foundational structure from Milestone 1. In this milestone, we implement **user interaction**, allowing players to flag cells, reveal them, and use rewards to peek safely — creating a fully **playable version of Minesweeper** in the console.

**🎯 Milestone Goal**

The goal of this milestone is to introduce **interactivity and game logic**, transforming the static Minesweeper board into a **functional game** where players make decisions, and the game responds accordingly.

**New Features in This Milestone:**

* Interactive cell reveal with **game over conditions**
* Ability to **flag** suspected bomb locations
* Collection and usage of **reward cells** for safe peeking
* Game **win/lose detection**
* **Input handling and validation** in the console

**🔑 Key Features**

* 🎮 **Interactive game loop** for real-time play
* 🏴 **Flag and unflag** cells to mark bomb locations
* 👁️ **Use rewards** to peek into unrevealed cells without risk
* 🎉 **Game win detection** when all safe cells are revealed
* ☠️ **Game over** if a bomb is revealed
* 📟 User-friendly **console prompts and messages**
* 🎨 **Console color output** to distinguish game states clearly
* 🧠 Improved logic for board updates, peeking, and status messages

**🗂️ Project Structure**

MineSweeperGame/

├── Cell.cs → Cell logic: bomb, visit, flag, reward state

├── Board.cs → Grid logic: reveal, flagging, bomb placement, game state

├── Program.cs → Main loop, input handling, player interaction

├── Enums.cs → Enum for game status (InProgress, Won, Lost)

├── Utilities.cs → Helper functions (input parsing, coordinate validation)

└── UML Diagram v2.png → Updated UML with game interaction

**🚀 Getting Started**

**✅ Requirements:**

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

**▶️ Run the Game:**

1. Clone or download the repository
2. Open the solution in Visual Studio or VS Code
3. Build the project to ensure everything compiles
4. Run Program.cs
5. Follow on-screen prompts to play the game interactively

**🧠 Summary**

This milestone completes the **core gameplay loop** for Minesweeper. The system now supports **user interaction**, **board updates**, and **game status checks** in real time.

Key C# concepts demonstrated in this milestone include:

* User input parsing and validation
* Control flow with switch, if/else, and while loops
* Enum usage for representing game states
* Encapsulation of logic inside class methods
* Interacting with 2D arrays dynamically
* Console UI enhancements for a better player experience

🎮 **Game Logic Highlights:**

* Flagging doesn't reveal a cell but marks it visually.
* Revealing a bomb ends the game immediately.
* Visiting a reward cell adds to the player's reward count.
* Using a reward allows one **free peek** into a hidden cell.

this as a downloadable README.md file or in GitHub-friendly Markdown format?