Problem Statement – 2048 Game Implementation (GUI Only)

Objective:
Develop a functional implementation of the popular game **2048** using any programming language of your choice. The goal is to combine tiles with the same number to reach **2048**. The game should have a **graphical user interface (GUI)** to display the board and allow the player to interact with the game.

Requirements:

1. **Board Initialization:**
    - Default board size: **4x4**.
    - Start with **two random tiles** with values 2 or 4.
2. **Game Mechanics:**
    - Players can slide tiles **up, down, left, or right** using keyboard or GUI controls.
    - Tiles with the same number **merge into one tile** with their sum.
    - After each move, a **new tile (2 or 4)** appears at a random empty position.
    - Game ends when either:
        - Player reaches **2048**, or
        - No more moves are possible.
3. **Technical Specifications:**
    - Board size should be **configurable (Y x Y)**.
    - Implementation should follow **functional programming principles**.
    - GUI should **update dynamically** after each move.
    - Track and display **score** based on merged tiles.
    - Support **game restart** from the GUI.

    - Code should be **modular, reusable, and readable**.

4. **Deliverables:**
    - Public **GitHub repository** with source code.
    - **Graphical user interface (GUI)** for gameplay.
    - **Deployed application link** (web-based GUI or packaged desktop app).
    - **README** explaining installation, running the game, gameplay instructions, and implementation details.

Interview Expectations:

- You **will be asked questions about your implementation**, including design, algorithms, and data structures.
- You will be asked to **change or add features in the game** during the interview.
- Be ready to discuss **functional programming principles**, **state management**, and **GUI updates**.