# Minesweeper Game
Systems Programming

1   Introduction

In this project, the aim is implementing a Minesweeper game. Your objective is to reveal all cells while avoiding mines. Each cell is numbered based on the total number of neighboring mines. For the game rules and strategies, you can refer to this link. Additionally, you can play Minesweeper online on this website to observe the game mechanics.
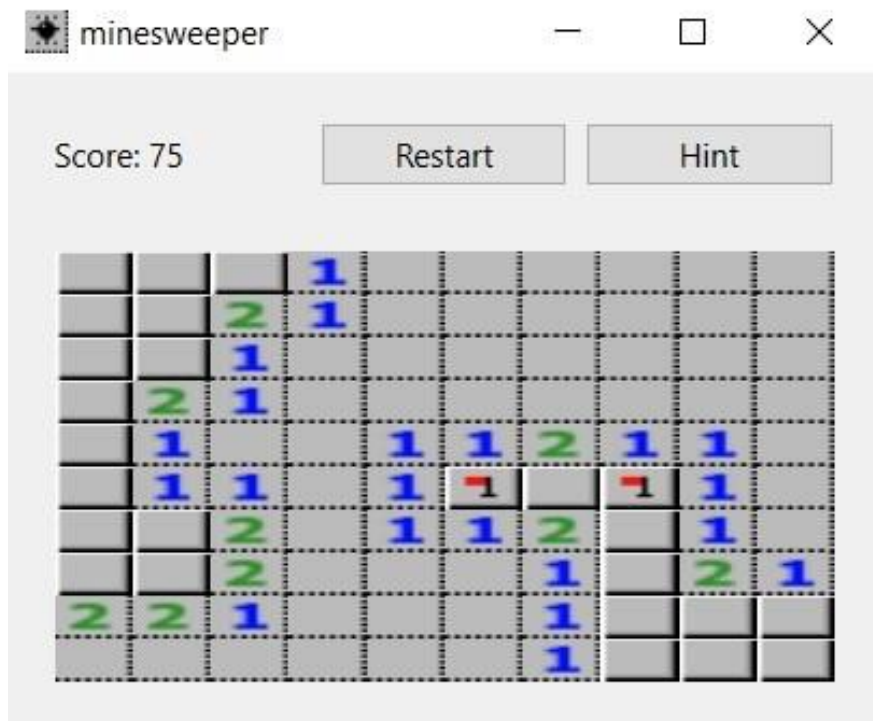
2   Functionalities

2.1   Game Mechanics

- Only unrevealed cells should be clickable (right or left-click).

- Left-clicking a cell should reveal either a number indicating neighboring mines, an empty cell, or a mine.

- Numbers should display the total number of neighboring mines.

- Cells with no neighboring mines should be revealed as empty.

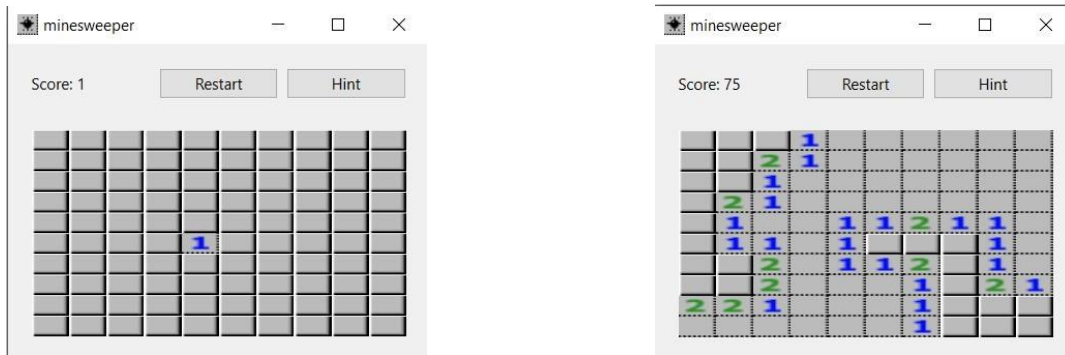- Revealing an empty cell should also reveal neighboring cells recursively.

Figure 1: We are clicking on the cell adjacent to the revealed '1'.

Figure 2: After we click, all neighboring cells are revealed recursively.

- Right-clicking should toggle the flag on unrevealed cells.

- Once revealed, cells must not be clickable (right or left-click).

- A restart button should initiate a new game with a different mine layout.

- A label should show the number of revealed cells as score.

- There should be hint button.

- Number of rows, columns, and mines in the game should be easily configurable with variables N, M and K. We will ask you to distribute K mines randomly in $N×M$ grid. For example, we may ask you to start the game with 10 mines in $10×10$ grid or another different configuration. So, be prepared.

2.2   End Game

- Left-clicking a mine is a losing condition.

- Revealing all cells without mines, is a win condition.

- After winning or losing the game, all mines must be revealed. A pop-up or notification should appear.

- The restart button must still be clickable after closing the pop-up.

- All cells must be unclickable until a new game starts.
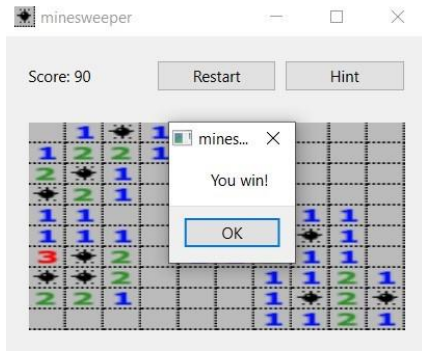
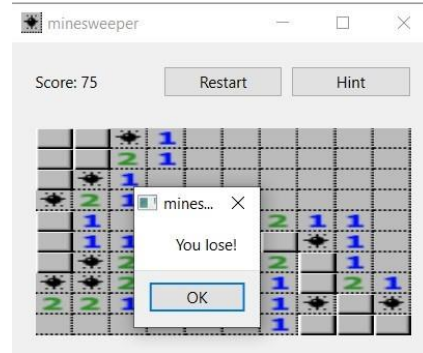Figure 3: Avoided all mines, won the game.

Figure 4: Clicked a mine, lost the game.

2.3   Hints

- Clicking the hint button should suggest an unrevealed cell that does not contain a mine.

- If the suggested cell hasn't been revealed by player, the second click should reveal it.
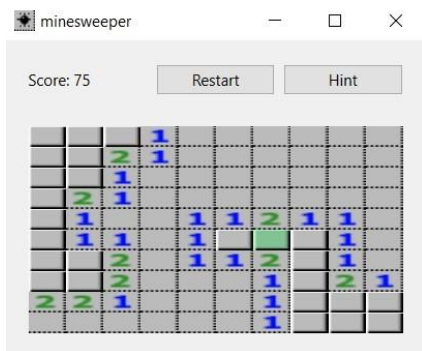

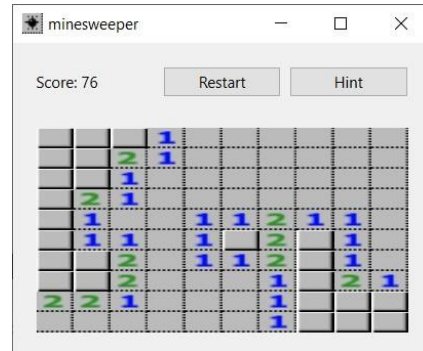Figure 5: First click on hint

Figure 6: Second click on hint

- The hint button should provide a hint based on the information visible to the player. Each hint should reveal a safe move, indicating a cell that is guaranteed not to contain a mine from the player's perspective.

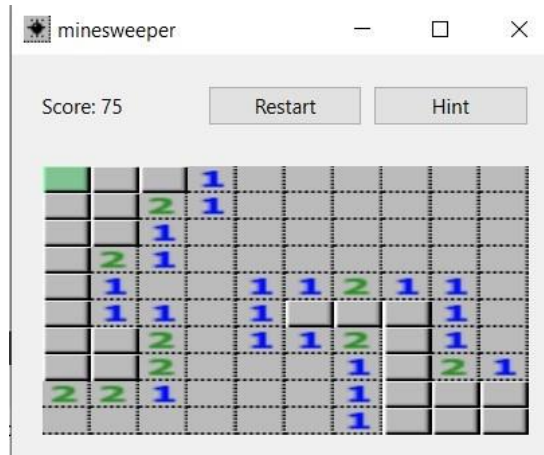- If no safe move exists, the hint button should not suggest any move.

Figure 7: This is an invalid hint because we don't know certainly whether there is a mine in that cell at the current state visible to player..