Zainab eman | 22f-3738

Submitted to: Sir Imran Baber

Minesweeper Game

Innovation Drive

OOP Course

2023

**Minesweeper Game**

The Minesweeper game is a classic single-player puzzle game where the player's objective is to clear a minefield without detonating any hidden mines. The game board is represented by a grid, and the player must strategically reveal cells while avoiding mines.

This documentation provides an overview of the Minesweeper game implementation, focusing on the key features, design, and functionality.

Features::

The Minesweeper game implementation includes the following features:

* **Object-oriented design:** The project is implemented using object-oriented programming concepts in C++.
* **Random mine placement:** Mines are randomly distributed throughout the game board, ensuring a unique experience with each game.
* **User interaction:** The player can input coordinates to reveal cells on the game board.
* **Game status:** The game tracks the player's progress and displays the outcome (win or lose) at the end of the game.

Implementation Details

The Minesweeper game is implemented using C++ and follows an object-oriented design approach. Here are the key implementation details:

Classes:

A class minesweeper in created in which two pointers are created, In Constructor both the pointers are given the address of 2D-Array dynamically using the concept of DMA.

Memory Management:

Dynamic memory allocation is used to create the game board, represented as a matrix of Cell objects. Memory is freed using a destructor in the Board class, ensuring proper cleanup after the game ends.

Unique Pointer type object:

In main function object is created dynamically using unique pointer to manage the memory and to prevent the complexities created due to out of bound. At the end of the game this unique pointer automatically call destructor thus freeing up space from heap.

**Flow of Execution**

The game follows the following execution flow:

1. The Game class is instantiated, and the 7 by 7 board size is prompted.
2. The Board is initialized and function (Minehere) place randomly places mines.
3. The game starts, and the Game class displays the initial game board.
4. The player is prompted to enter coordinates to reveal a cell.
5. The Board class updates the game board based on the player's input.
6. If the revealed cell contains a mine, the game ends with a loss.
7. If the revealed cell does not contain a mine, the game continues until either all safe cells are revealed (win) or a mine is revealed (loss).
8. The game status and final score is displayed, and exit.

Conclusion:

The Minesweeper game implementation provides an enjoyable and challenging gaming experience. It showcases object-oriented