

# Applications

---

Tai, Wei Hsuan  
week 9

# Announcement

---

Recording or photographing the slides or any content displayed on the screen is permitted for personal use only.

Redistribution, modification, or any commercial use of the materials is strictly prohibited.

© 2025 [Tai, Wei Hsuan]. All rights reserved. For personal use only.

# Course Materials



# Outline

---

## 1. Minesweeper

Overview

Analysis

Let's Start!

# Minesweeper

---

# Outline

---

## 1. Minesweeper

Overview

Analysis

Let's Start!

## What is Minesweeper

Minesweeper is a classic single-player puzzle game that challenges players to uncover a grid of cells while avoiding hidden mines. The objective is to clear the board without detonating any mines, using numerical clues provided by revealed cells to deduce the locations of the mines.

# Outline

---

## 1. Minesweeper

Overview

Analysis

Let's Start!

## Gaming Process

1. The game starts with a grid of covered cells, some of which contain hidden mines.
2. Players click on cells to reveal them. If a cell contains a mine, the game ends.
3. If a cell does not contain a mine, it reveals a number indicating how many adjacent cells contain mines.
4. Players use these numerical clues to deduce the locations of the mines and avoid clicking on them.
5. Repeat the process until all non-mine cells are revealed, resulting in a win.

## What do we need?

---

- A grid to represent the game board.
- Random placement of mines within the grid.
- Logic to calculate and display the numbers indicating adjacent mines.
- User input handling for cell clicks.
- Game state management to track wins and losses.

# Outline

---

## 1. Minesweeper

Overview

Analysis

Let's Start!

## Generate the Game Board

In this phase, player should enter the size of the grid and the program would automatically generate the game board with mines randomly placed. Remember to ensure the input size is within the acceptable range (e.g., 5 to 10).

Refer to `random.cpp` in the google classroom for random number generation.

## Check the result

After generating the game board, we can display the answer key to verify the correct placement of mines and the corresponding numbers for each cell. This step is crucial for debugging and ensuring the game's logic is functioning as intended.

## Showing the map to players

Finally, we need to implement the functionality to display the game board to players. This involves revealing the cells based on player input while keeping the mines hidden until the game ends. Players should be able to see the numbers indicating adjacent mines as they uncover safe cells.

## Start minesweeping

Note that in the CLI (Command Line Interface), players would need to input the coordinates of the cell they want to reveal. The program should handle this input, update the game board accordingly, and check for win/loss conditions after each move.

After they input the coordinates, update the `record` array to mark the cell as revealed and call the `show_map()` function to display the updated game board.