

WETHINKCODE_

BOOTCAMP

GROUP PROJECT I

Minesweeper

Boot Camp 2021



CONTENTS

1	Minesweeper	2
2	Introduction	2
3	Objectives	3
4	Instructions	3
5	Mandatory	5
5.1	Level 1	5
5.2	Level 2	5
5.3	Level 3	5
5.4	Bonus	6
6	Plagiarism Declaration	6

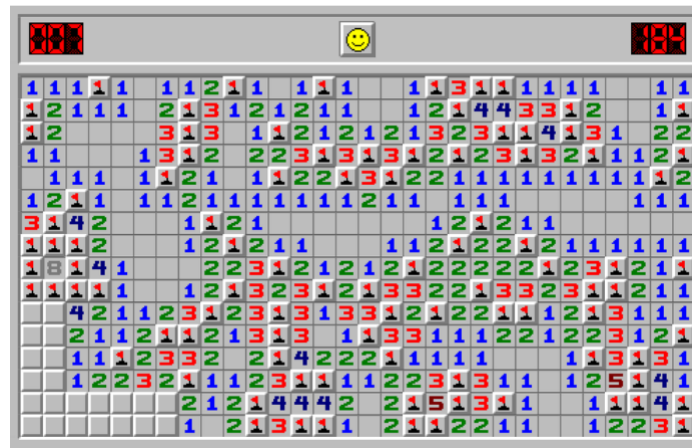


Figure 1: Minesweeper by Tornad00910 on [WikiMedia](#)

1 MINESWEEPER

Congratulations! You have made it to the final project for the WeThinkCode_ Bootcamp.

Your next task is to take on a group project.

Your project is "Minesweeper", to find out more about this game or to play it a little bit for "research" purposes go to:

[MinesweeperOnline.com](#)

2 INTRODUCTION

The most successful game ever made isn't World of Warcraft, Tetris or even the Sims

Click-for-click, nothing has wasted more time than Minesweeper, a little game that debuted back in 1990 as part of the Windows Entertainment Pack, before being promoted to a standard feature in Windows 3.1 and onwards.

Minesweeper was a Microsoft original, written by Robert Donner and Curt Johnson, and hasn't changed much over the years.

In the unlikely event you've never played it, the gist is that you start with an empty field (its size and number of mines determined by difficulty setting) and have to uncover squares one at a time. Underneath each is either a space, a number, or a mine. The numbers tell you how many mines are in the adjacent boxes, the mines kill you dead. To win, you have to clear the field without touching a mine. It's a relatively simple game of deduction, but satisfying.

Source: [Tech Radar](#)

3 OBJECTIVES

The idea is to get you to see the possibilities with Python and the Command Line. You have gone through all the exercises and now we must determine if you can apply all the concepts with your own Problem Solving skills.

Therefore, we wish to remind you, this is **NOT a MATH Problem** but rather a Problem Solving question.

4 INSTRUCTIONS

- You are expected to design a function or multiple functions that will recreate the famed game Minesweeper.
- The board for the game must be taken in as input. The board must be square with validations to ensure its dimensions cannot be less than size 16 i.e 4x4.
- The number of bombs must be taken in as input and cannot exceed the total dimensions of the board i.e bombs < squares
- The placement of the bombs should be random on every new iteration of gameplay. (When a new game starts, don't move the bombs when people are playing)
- The game must display when the User has Won! This happens when all the squares have been opened without detonating a bomb.
- The game may look in any way that you wish but our recommendation is shown in Figure 2 on the following page (easiest to implement)
- You may take input in any method you wish but we would recommend a row & column system as shown in Figure 3 on the next page.

Good Luck!

```

y
  0 1 2 3 4 5 6 7 8 9
-----
0 | | | | | | | | | |
1 | | | | | | | | | |
2 | | | | | | | | | |
3 | | | | | | | | | |
4 | | | | | | | | | |
5 | | | | | | | | | |
6 | | | | | | | | | |
7 | | | | | | | | | |
8 | | | | | | | | | |
9 | | | | | | | | | |
-----
Input as row,col a block to search: █

```

Figure 2: Minesweeper print(board) function call

```

Input as row,col a block to search: 5,5
  0 1 2 3 4 5 6 7 8 9
-----
0 | | | | | | | | | |
1 | | | | | | | | | |
2 | | | | | | 2 1 2 | |
3 | | | | | 1 0 1 | |
4 | | | 1 1 1 1 0 1 | |
5 | | | 1 0 0 0 0 1 | |
6 | | 2 1 0 0 0 0 1 | |
7 | 2 1 0 0 0 0 0 1 1 |
8 | 1 1 0 0 0 1 1 1 0 0 |
9 | 0 0 0 0 0 1 | 1 0 0 |
-----
Input as row,col a block to search: █

```

Figure 3: Minesweeper Row,Col System (5,5)

5 MANDATORY

You will have to answer the questions according to levels. You need to complete one level before proceeding to the next level. Each level is designed to test concepts you have learnt throughout the Bootcamp.

5.1 Level 1

In order to complete this level you will need to complete the tasks listed below:

1. Print a board based on input.
 - The input will be in X, Y format
 - X is the number of bombs
 - Y is the size of one side of the board
 - $X > 0$ and $X < Y$
2. The board must show where the bombs are placed
3. The board must randomize where bombs are placed each time a game is run
4. There must always be at least one bomb
5. A game cannot consist entirely of bombs

5.2 Level 2

In order to complete this level, you must retain all the functionality of Level 1 and add this feature:

1. In the squares that do not have bombs, display a number in that square. This number is equal to the number of bombs around that square both above, below and diagonally.
2. You are still expected to show the bombs.

5.3 Level 3

In order to complete the third level, you must alter programme in the following ways:

1. Let the console have two options:
 - 'Play Game' - Level 3 Requirements
 - 'Show Solution' - Level 1 & 2 Requirements
 - 'Exit' - Quit the game

2. Hide all the bombs and all the numbers from Level 2
3. Take input in a row & column set to reveal the reveal the square contents.
4. All conditions listed in [Instructions](#)

5.4 Bonus

Do *NOT* attempt a Bonus unless all three (3) levels have been completed fully.

- After a Winning or Losing condition, allow a player to return to a Welcome Screen offering an opportunity to try again.
- Create a Scoreboard system that can be viewed from the game.
- Let the scoreboard system be written to a file and loaded on each gameplay with a Player Name and Score.

6 PLAGIARISM DECLARATION

Engaging in any cheating or dishonesty in any form of assessment, assignment, test or examination or other WeThinkCode_ prescribed work is considered cheating and is grounds for disciplinary action. Plagiarism, which is to present work (or a portion of work) as your own when it is not, is considered cheating and is not accepted at WeThinkCode_.

An evaluator can flag one for plagiarism on one of the following grounds :

- The evaluator (marker) identifies that the student does not understand all or part of the work they have submitted.
- If all or part of the work presented is plagiarised ,i.e. copied from another source without reference.

Cheating in group projects

The main purpose for a group project is to give students the experience of working in a team, by coming up with a solution to a problem together.

- Each member must be able to show which portion of the project they worked on.
- Failure to do so will result in the student being flagged for cheating which will be grounds for disciplinary action.

- This is to avoid single members doing the majority of the group project at the benefit of a member who is not contributing.
- In this way we are able to ensure fair assessment of each WTC_student's competence.

Group projects can be approached in two ways.

1. Divide and conquer: This is usually preferred and advised when working on big projects. The project is divided into segments, in which each member of the group can accomplish. Once completed, the group will then integrate the segments to complete the project
2. One for all: This method is usually preferred and advised when a group is working on a small project. The group will work on the solution together from the start of the project until the end. This will require the members to move at a pace in which everyone in the team can keep up with.

NOTE: At the end of each group project, each member should have a general and basic understanding of the project and the solution found. This will include running, testing and explaining the solutions of the project.

DECLARATION

I hereby declare that the work submitted by me and/or my group members is:

- Original (not plagiarised)
- References listed
- Honest & in Good Faith
- Subject to WeThinkCode_policies

(YOUR NAME)

WeThinkCode_ Bootcamper