code.sprint MT

TASK BOOKLET

- Final Round -

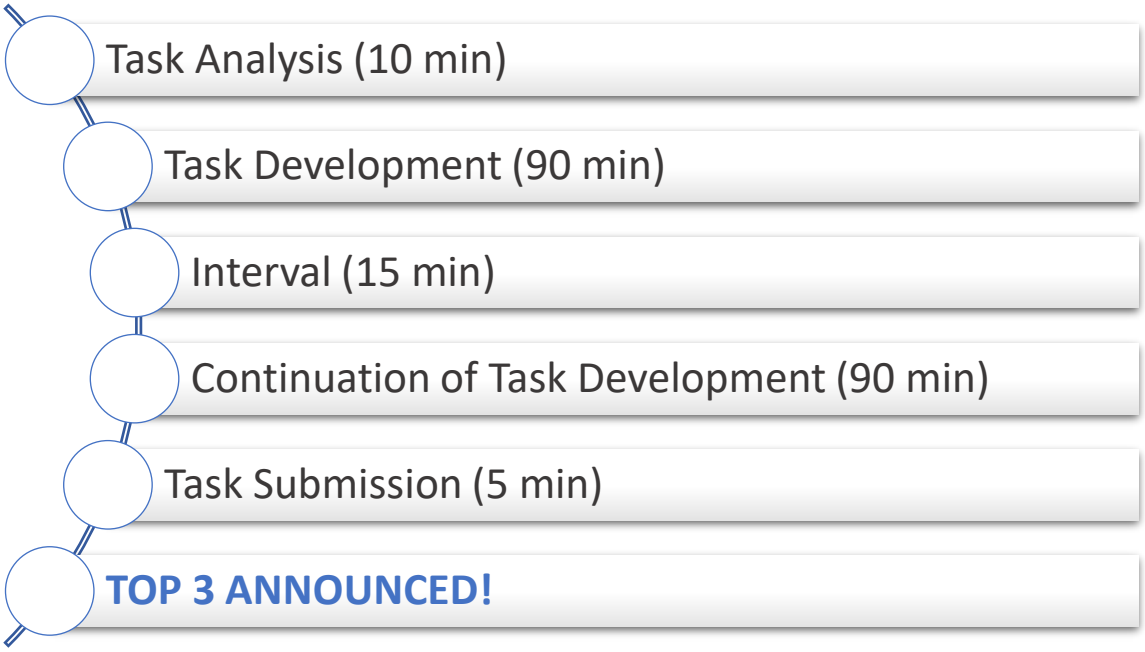
Post-Secondary Category

2021

DIRECTORATE FOR LEARNING & ASSESSMENT PROGRAMMES

x ICE Malta

Finals Round Schedule



Bombsweeper (180 minutes)

Bombsweeper is a single-player puzzle game. Its objective is to clear a cell-matrixed rectangular board containing hidden bombs without detonating any of them, with help from clues about the number of neighbouring bombs in each cell. The game originates from the 1960s, and since then various variations and offshoots have been developed.

Develop a text-based Bombsweeper whereby the player must locate the bombs without detonating any of them. Each cell, when selected, will output how many bombs are next to it …but if it's a bomb, it will explode! A cell can also be empty if there are no adjacent bombs.

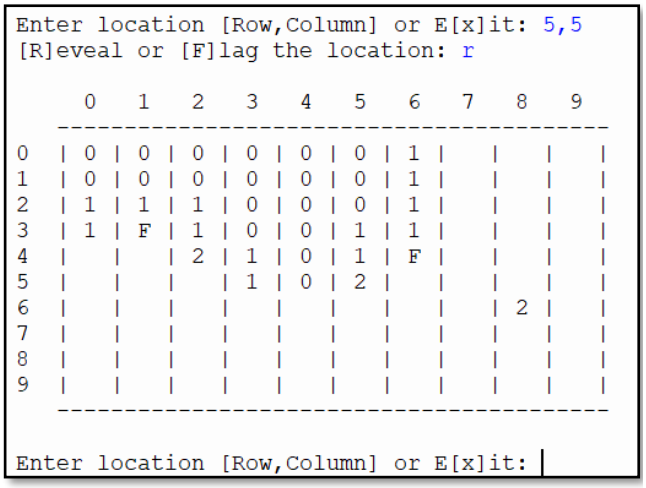
Functionality #1: Gameplay

1. The board is a 10 x 10 cell matrix (0 - 9 rows and 0 - 9 columns).
2. The player can choose a cell by indicating the row and column or exit the game. Check sample screenshot 1 below.

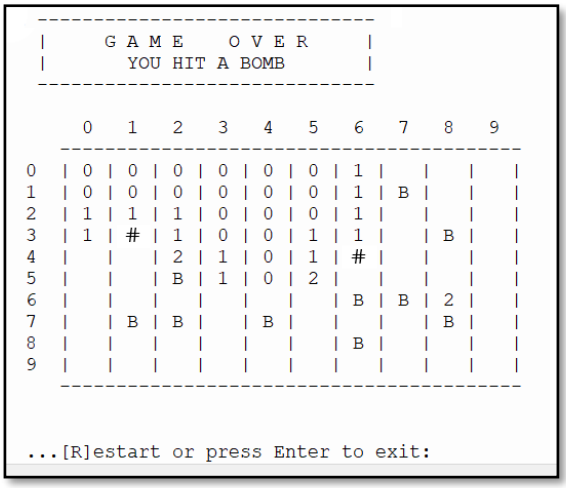
```
    0   1   2   3   4   5   6   7   8   9
  -----------------------------------------
0 |   |   |   |   |   |   |   |   |   |   |
1 |   |   |   |   |   |   |   |   |   |   |
2 |   |   |   |   |   |   |   |   |   |   |
3 |   |   |   |   |   |   |   |   |   |   |
4 |   |   |   |   |   |   |   |   |   |   |
5 |   |   |   |   |   |   |   |   |   |   |
6 |   |   |   |   |   |   |   |   |   |   |
7 |   |   |   |   |   |   |   |   |   |   |
8 |   |   |   |   |   |   |   |   |   |   |
9 |   |   |   |   |   |   |   |   |   |   |
  -----------------------------------------

Enter location [Row,Column] or E[x]it:
```

Screenshot 1: Sample board matrix

3. Upon game initialisation, ten bombs (B) are placed in random cells along the board.
4. At every turn the user must indicate whether to:
   - Flag (F) a cell to indicate it is a bomb, or
   - Reveal (R) the cell's contents and depending on its content, the system will display: Bomb (B), or Number of bombs adjacent to that field (0 – 8).
5. If a chosen cell is already flagged, the user can unflag it or leave it flagged.
6. The user does not have access to an already revealed cell.
7. At every turn, the updated board should be displayed, such as in the sample screenshot 2 below.
8. The winning point is when the player sweeps all bombs from the field by correctly flagging them.
9. The player has only one life. If the player reveals a bomb-containing cell, the bomb detonates, and the game is over.
10. At game over, besides showing the user's moves, the board should also reveal the location of all the bombs. Correctly flagged bombs should be display with a '#' sign. The user is also given the chance to restart or quit the game. Check sample screenshot 3 below.



Screenshot 2: Updated board matrix



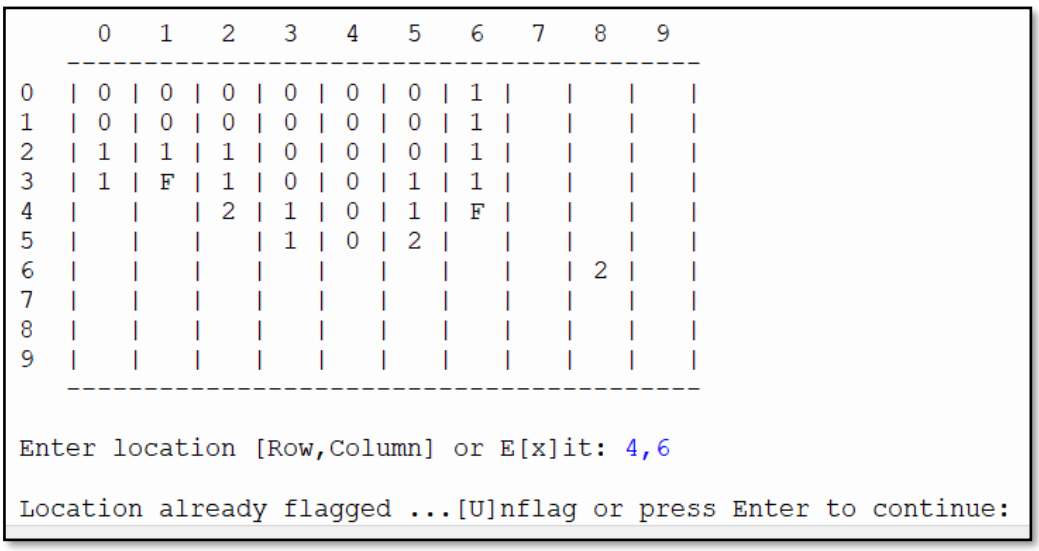
Screenshot 3: Game Over

Functionality #2: Inputting and Validation

1. The player must first enter the location. The location should be in the form:
   - Row , Column (such as 3,6); OR
   - Letter 'X' to quit the game.

2. Afterwards, the player can then: 1) press 'R' to reveal the cell, or 2) press 'F' to flag the cell, or 3) press 'U' to unflag the cell in case it is already flagged (check sample screenshot 4 below).

3. All data inputting is not case sensitive and proper messages must be displayed in case of invalid entries.



Screenshot 4: Already flagged location

Name the class containing the main method **RunApp.**
Submit your program in a folder named **BombSweeper**

Assessment Rubric

Program Functionality	User-Friendly Interface	Proper use of Comments	Proper Conventions (Camel case, meaningful var names etc.)	Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed
Generating Random Bombs Locations	Calculating the number of neighbouring bombs	Validation of Location (Row & Column or X)	Validation of Action (Reveal, Flag, Unflag)	Proper use of data structures and/or files	Modular Code	Code Efficiency
Properly updating the status of the board locations with every user's turn		Properly reveal location of all bombs and user inputs at game over			Other Features (Not listed in the task)	

Maximum Score: 32
+ 2 for every extra feature

0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily


eSkills
eSkills Malta Foundation


HSBC | Malta Foundation


melita foundation


Tech.mt
MALTA - LEADING THROUGH INNOVATION


FreeHour