# Problem Set 3 Exercise #30: Minesweeper [Challenge]

**Reference:** Lecture 9 notes

**Learning objectives:** Two-dimensional array; Char array; Algorithm design

**Estimated completion time:** 180 minutes

## Problem statement:

*Minesweeper* is a computer game whose objective for the player is to clear a minefield without detonating a mine. The player is presented with a grid of squares (Figure 1) under which some contain mines, and some do not.

For a square that is safe (mine-free), it contains an integer value (0 to 8) indicating the number of mines surrounding it.

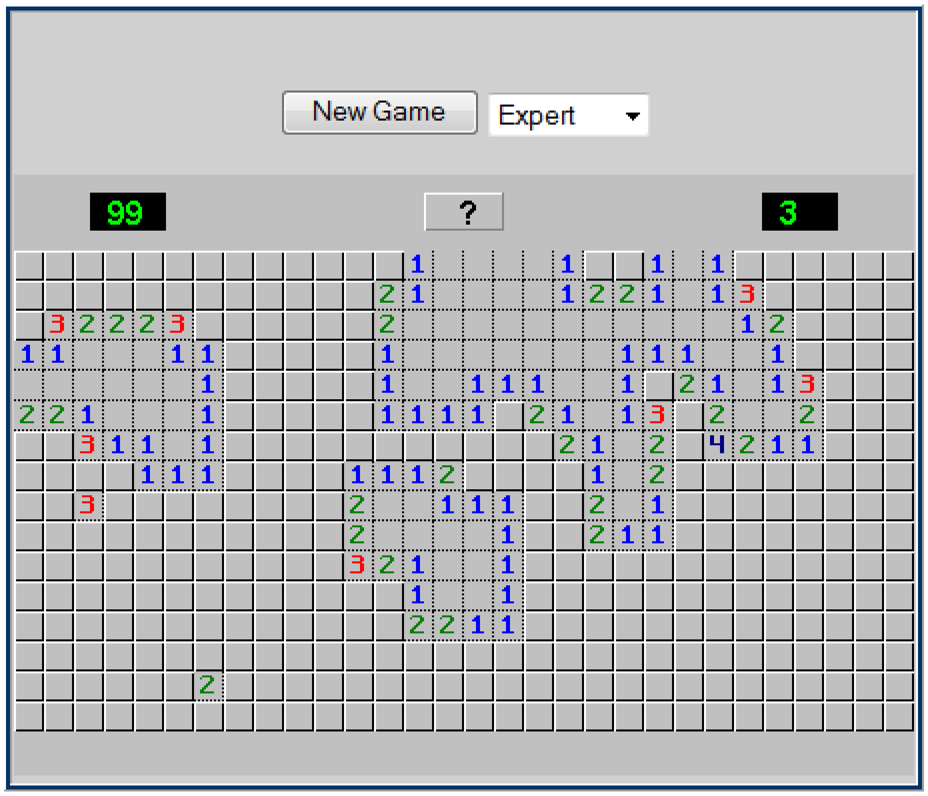


**Figure 1.** A Minesweeper game in progress

A player clicks on a square to turn it over. If it contains a mine, the game ends and all the mines revealed (Figure 2). The game also ends when the player turned over all safe squares. Usually, a square that contains 0 (no mine around it) is simply displayed as a blank square.

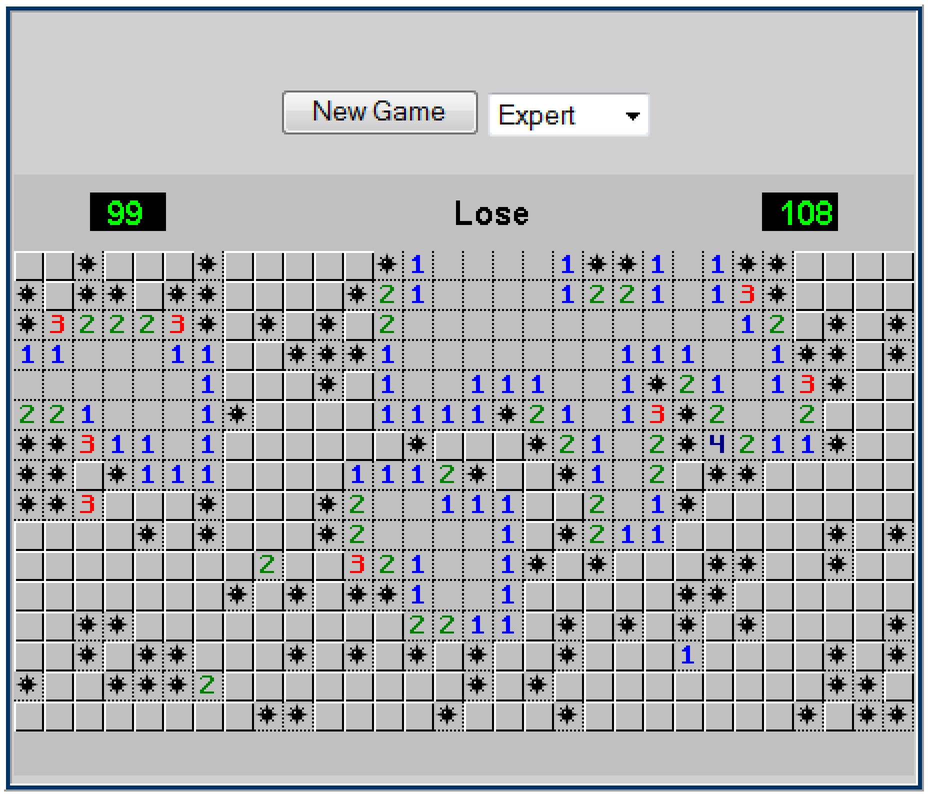
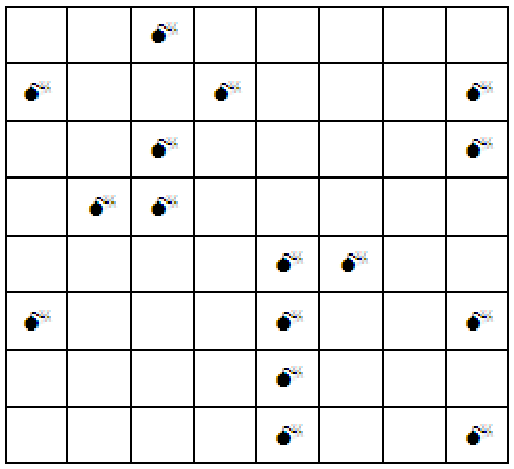


**Figure 2.** The Minesweeper game ends when player clicks on a square containing a mine.

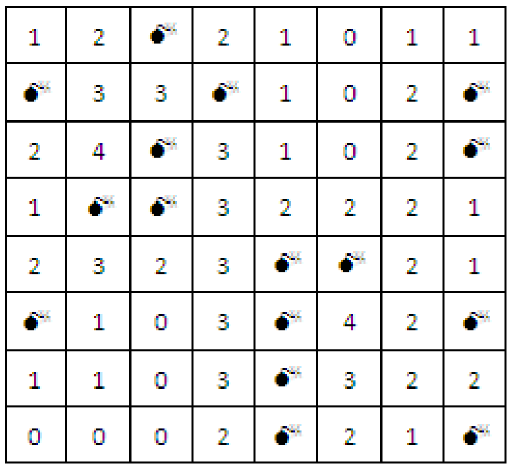
For this exercise, you are only required to prepare the grid before the start of the game. Given the positions of all the mines in the grid, you are to fill in the numbers 0 to 8 in each of the safe squares and 9 in a square with mine.

For example, Figure 3 shows the positions of the mines in a minefield. You are to compute the values of the safe squares, as shown in Figure 4.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | ● | | | | | |
| ● | | | ● | | | | ● |
| | | ● | | | | | ● |
| | ● | ● | | | | | |
| | | | | ● | ● | | |
| ● | | | | ● | | | ● |
| | | | | ● | | | |
| | | | | ● | | | ● |

**Figure 3.** Positions of mines



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | ● | 2 | 1 | 0 | 1 | 1 |
| ● | 3 | 3 | ● | 1 | 0 | 2 | ● |
| 2 | 4 | ● | 3 | 1 | 0 | 2 | ● |
| 1 | ● | ● | 3 | 2 | 2 | 2 | 1 |
| 2 | 3 | 2 | 3 | ● | ● | 2 | 1 |
| ● | 1 | 0 | 3 | ● | 4 | 2 | ● |
| 1 | 1 | 0 | 3 | ● | 3 | 2 | 2 |
| 0 | 0 | 0 | 2 | ● | 2 | 1 | ● |

**Figure 4.** Values in safe squares

Write a program **PS3_Ex30_Minesweeper.java** to read in a minefield containing mines, and compute the values of the safe squares in a numeric 2-dimensional array.

There are 3 game levels, with each having the following grid dimension:

- Level 1: 8 × 8 grid
- Level 2: 12 × 16 grid
- Level 3: 16 × 30 grid

Your program is to read the game level, and then the grid containing characters comprising either **-** (mine-free) or ***** (a mine). Your program then outputs a 2-dimensional integer array showing the number of mines surrounding each square. If the square in the grid contains a mine, then its corresponding value in the integer array is 9.

**Sample run #1**:

```
Enter level (1 - 3): 1
Enter the 8 x 8 board:
--*-----
*--*---*
--*----*
-**-----
----**--
*---*--*
----*---
----*--*
1 2 9 2 1 0 1 1
9 3 3 9 1 0 2 9
2 4 9 3 1 0 2 9
1 9 9 3 2 2 2 1
2 3 2 3 9 9 2 1
9 1 0 3 9 4 2 9
1 1 0 3 9 3 2 2
0 0 0 2 9 2 1 9
```

## Sample run #2:

```
Enter level (1 - 3): 2
Enter the 12 x 16 board:
 --*-----*--*---*
--*----*-**-----
----**--*---*--*
----*-------*--*
**-----*---***--
*----*----*---*-
*--**---------**
--*--*-*------**
------***-------
---*-*-----***--
*---*----------*
---*-----*-----*
0 2 9 2 0 0 1 2 9 3 3 9 1 0 1 9
0 2 9 3 2 2 2 9 4 9 9 3 2 1 2 2
0 1 1 3 9 9 2 2 9 3 2 3 9 2 2 9
2 2 1 2 9 3 2 2 2 1 1 4 9 4 3 9
9 9 1 1 2 2 2 9 1 1 2 9 9 9 3 2
9 4 2 2 3 9 2 1 1 1 9 3 3 4 9 3
9 3 2 9 9 3 3 1 1 1 1 1 0 3 9 9
1 2 9 3 3 9 4 9 3 1 0 0 0 2 9 9
0 1 2 2 3 3 9 9 9 1 1 2 3 3 3 2
1 1 1 9 3 9 3 3 2 1 1 9 9 9 2 1
9 1 2 3 9 2 1 0 1 1 2 2 3 2 3 9
1 1 1 9 2 1 0 0 1 9 1 0 0 0 2 9
```

## Useful tips:

You might think that you need to deal with “inner cells”, “boundary cells” and “corner cells” separately. Actually that could be avoided by careful design.