



MineSweeper

JavaScript,html,SASS

Anan Aghbar



Contents

Setup instructions:	Error! Bookmark not defined.
Running the assignment:	2
Database Schema:.....	Error! Bookmark not defined.

Table of figures

Figure 1: Desktop application view.....	4
Figure 2: Mobile application view	4
Figure 3: Database Schema.....	5

File Structure:

- Js:
 - Minesweeper.js
 - Board.js
 - Cell.js
- Css
 - Minesweeper.sass
 - The compiled css file
- Index.html

Start Game:

To build a new game the following objects and methods are used:

- New instance from the Minesweeper object in the index.html page
- Minesweeper object has the following properties and functions:
 - Properties:
 - `components:`
 - `board:div.board`
 - `col:input.col`
 - `mine:div.mines-count`
 - `mines:input.mines`
 - `restartButton:button.restart-button`
 - `row:input.row`
 - `startButton:button.start`
 - `time:div.time`
 -
 - `dimension`
 - `initialize`
 - `isMinesweeperOver`
 - `leftMineCount`
 - `mineCount`
 - `time:`
 - Functions:
 - `MinesweeperIsover:f (isWin):` to check if the game is over or not.
 - `decrementLeftMineCount:f ():` to decrement mine counter by 1.
 - `increaseLeftMineCount:f ():` to increment mine counter by 1
 - `eventsHandlers:f ():` to assign the events to their handlers functions
 - `findCellByEvent:f (event):` return Cell object of the clicked cell.
 - `generateGame:f (rows,cols,mines):` assign the game dimensions and mines
 - `init:f ():` Initialize the Minesweeper object's elements and properties
 - `isWin:f ():` check the player is win or not
 - `leftClickHandler:f (event):` to process the clicked cell
 - `restartClickHandler:f ():` to rest the game
 - `rightClickHandler:f (event):` to flag/unflag the suspected mine
 - `startClickHandler:f ():` to start new game button
 - `startTimer:f ():` enable the timer
 - `stopTimer:f ():` disable the timer
 - Board object.

- Board object has the following properties and functions:
 - Properties:
 - `cells:Array` Cells objects (game cells)
 - `cols.`
 - `rows.`
 - `dimension`
 - `element:div.board`
 - `mineCount.`
 - Functions:
 - `appendClearfixElement:f ()`: to append clear fix element between each column and row.
 - `calculate:f ()`: to calculate the number of the siblings for each Cell.
 - `draw:f ()`: to draw the cells and append the cells elements
 - `getCellSiblings:f (cell)`: to get the siblings of the cell
 - `getNotOpenedCells:f ()`: to get the unopen cells
 - `getRandomNumbersArr:f (maxRow)`: to get an array of random numbers
 - `init:f (rows,cols,dimension, mineCount)`: to initialize the board object
 - `open:f ()`: to open the board's cells
 - `plantMines:f ()`: to distribute the mines on the board's cells randomly.
- Cell object has the following properties and functions:
 - Properties:
 - `element:span.cell`
 - `isEmpty.`
 - `isFlagged.`
 - `isMine.`
 - `isOpened.`
 - `mineCount:0.`
 - `x.`
 - `y.`
 - Functions:
 - `open:f ()`: open the clicked cell
 - `setEmpty:f ()`: set the clicked cell as empty (has no mine)
 - `setFlagged:f ()`: set the clicked cell as suspected mine
 - `setMine:f ()`: set a mine on the current cell
 - `setMineCount:f (number)`: set the mine count
 - `setOpened:f ()`: set the open flag.

Game Screen Shots:

MineSweeper Assignment in java script

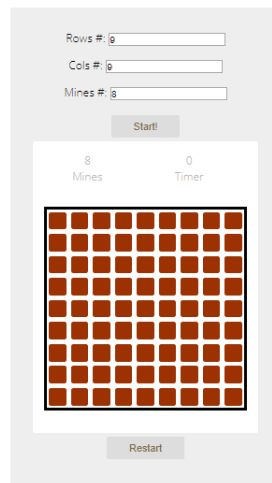


Figure 1: New Game

MineSweeper Assignment in java script

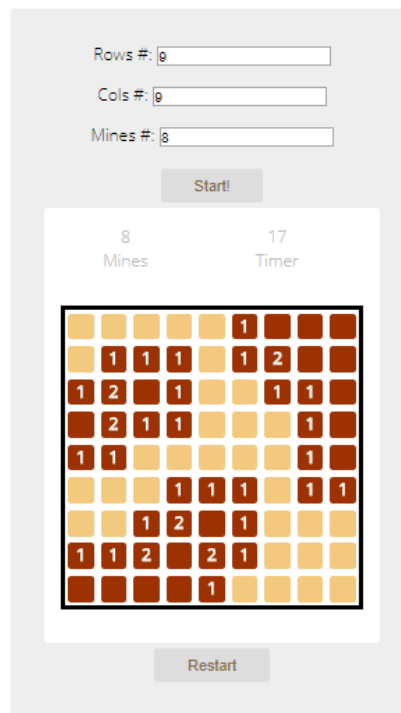


Figure 2: During the game

MineSweeper Assignment in java script

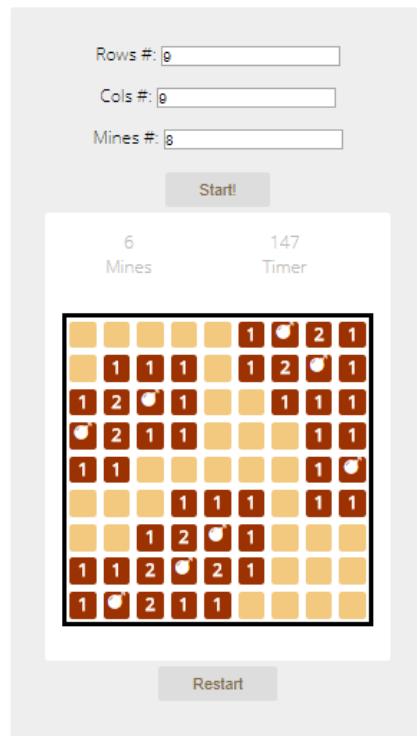


Figure 3: Game Over