



T - Web Development Seminar

T-WEB-x00

2048 / mindMap

Back To Javascript





2048 / mindMap

repository name: web_seminar_day13_\$ACADEMICYEAR
repository rights: ramassage-tek
language: javascript



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).



All requested methods in the following exercises must be executed after complete loading of the page.

There are 2 subjects.

You are free to choose the one you prefer, or to do both:

- **2048**: it consists in a **2048 game** in jQuery.
- **mindMap**: it consists in a **mind map application** in jQuery.

Your project must be encapsulated in a plugin that makes it easy to integrate within an HTML page.



For all the exercises, you will need the code of the preceding exercise to integrate the requested new functionalities.

In each case, the file to hand in must be named *main.js*.



Most of the instructions are not strict but indicative. If you think you have a better idea, you can try it.



EXERCISE 01

2PTS

Restriction: the use of `$.fn`

Use the functionalities of `$.fn.extend` to create an empty plugin named `mygame` or `mymindmap` with a function having a size as parameter.



Check [this](#).

EXERCISE 02

2PTS

Within this function, create the following variables:

- `gameObject` or `mapObject` that gets the id of the element to which the plugin is applied.
- `blockSize` or `mindSize` that records the size of the side of an atomic element (whether a square passed or an atomic element of the map, that we'll call *mind*).
- `boardSize` or `mapSize` which represents the size of the side of the game board or the map (both begin square).



To calculate the board size, take into account that the game board is a square in which are laid out 16 small identical squares, for which the length of one side is worth `blockSize`.

Once these 3 variables are created, call a method that introduces a div element in the element calling the plugin.

This div must have as id *board* or *mind-master* and as size the value specified by `boardSize` or `mapSize`.



You also can use a class "board" or "mind-master" if you want to make a more generic plugin.



EXERCISE 03

4PTS

+ 2048

Our board is for now virtually divided into 16 sub squares.
Implement now a function inside your plugin which inserts 16 div in the parent div *board*.



All these div will have to fill the board div and make up a 4 x 4 div square.

These div will have **blockSize** as size length.

We will add the class **square-container** to those div.
To display it properly, let's add a margin to those div.
Finally, let's order the squares properly. The goal is to obtain a 4x4 grid.



Use CSS (inside our plugin) for that...

+ MINDMAP

Our board now contains a *mind-master* div, which will be linked with all new minds.
Implement a function inside your plugin which inserts one new *mind* div to the parent div *mind-master*.
This function has to take one string parameter that contains the text to display in the mind.
These div will have a width based on "mindSize".

Let's add the class **mind-container** to those div (you can add a div inside to store the text if you want to).
Add a style sheet to these divs.
Finally, let's order the squares properly. The goal is to obtain a readable mind map.



The minds must be linked to the master.
Theses links must be visible.
The minds should surround the master, but you are free to place them the way you want.



You can draw the lines with canvas or CSS.



EXERCISE 04

2PTS

+ 2048

Now we have a grid, let's create a function that randomly adds a tile to the grid.
The placement of the tile must be random among empty cells.
The value of the generated tile must be random between 2 and 4.



The first time this function is called, 2 tiles must be created at the same time.

+ MINDMAP

We now have a map, and can create new minds.
The next step is to have the user to modify the minds himself.
Bind the double left click to allow the user to edit the text of the div, changing it in an editable text field.
When he's done and click left again somewhere else, the div should go back to its immutable state.



There is an event to do this.



EXERCISE 05

5PTS

+ 2048

Let's implement the movement of tiles.

You need to handle the four directions: when a arrow key is pressed (left, right, bottom, top), all tiles must move toward the direction of the key in the grid as far as they can go.

Moreover, every time a key is pressed a new tile must be generated.

You may choose yourself if you want those actions to happen as soon as the key is pressed or after the key has been released.



One is easier than the other.



It is more difficult than it seems.

+ MINDMAP

Let's implement the addition of sub-minds.

If the user clicks on a mind, and then press "Enter", it should create a new mind, linked to the selected one.



You should also be able to create a new sub-mind to a sub-mind...



You should just improve your function from exercise 03 with a new argument "parent".



EXERCISE 06

2PTS

+ 2048

The most important step is still missing: when a tile collides with another, they must melt if they have the same value.



When more than one collision happens at the same time in the same row / or column, the priority is given to the tiles which are closer to the side where the movement is directed.

For example:

```
0 0 0 0
0 2 2 2
0 0 0 0
0 0 0 0
```

gives, by moving right:

```
0 0 0 0
0 0 2 4
0 0 0 0
0 0 0 0
```

+ MINDMAP

The last step is now to have the minds movable.

That means the user has to be able to drop-down any mind (except the master) and change its position (under / over another) and its parent:

- if the user puts a mind on another one, it should add it as its child,
- if he moves it elsewhere, you should just make it move it around its parent.

EXERCISE 07

3PTS

Your project should now be completed, but plain looking and not very attracting.
To make it nice:

- use CSS or add some images to represent the elements,
- add some jQuery animations.
- make the elements expandable and reducible.



BONUS

You've done with everything?

Congratulations!

Hope you also checked that your code is clean and follow the jQuery plugin good practices...

Here are a few ideas as bonus for each project.

+ 2048

- Multiple skins of images which can be selected in a dropdown list (a skin Star Wars would be great!),
- a page and a grid more beautiful,
- undo button,
- score,
- skin of image directly "uploaded" from your webpage (i.e images can be specified manually for each tiles)...

+ MINDMAP

- Create a new kind of links between any mind,
- add options to change the colors, fonts,...,
- make the font adapted to the div size,
- build a collaborative application. Every user can modify the same mindmap and see the other's modifications.