

## Assessment02: Evaluation

### **Goals and Implementation**

We managed to implement all our project aims and kept to our original layout. We thought of how we should go about coding and came up with a brief outline – separate JavaScript files each for grid handling, user interface event handling, display update, the model and the game animation loops. One HTML file for the jQuery mobile display, one CSS file to add a bit of CSS on top of the jQuery CSS library.

We first started by laying down the system model, the variables and how the grid would work, we settled on a 2D array with each variable representing a box with a type (colour), and each of these boxes would be a certain pixel size to fit perfectly in the canvas, the canvas size is 256px \* 256px. Our grid is 8 by 8 so each box is 32px \* 32px and contains a circle of the colour it was attributed to randomly.

We then started to build the event handling for mouse controls and the game loops. The game works as follows:

Every 200 milliseconds, it checks the whole grid for correct alignments and marks boxes that need to be removed, then the boxes get removed and adds up to the score before putting new ones in, or making the one fall into empty slots. The game then creates new random boxes from the top.

Lastly we developed the display handling. It includes the function that draws everything in the canvas according to the game state. We also added in the top scores (top 15) and the functionality to change the language of the instructions.

For the difficulty setting, there were multiple ways of doing it (unmovable blocks, poisoned, boxes, etc.) What we ended up doing was the harder difficulty the user chose, the more colours there were. It therefore became more complicated to make alignments.

Easy – 6 different colours

Medium – 7 different colours

Hard – 8 different colours

### **Key Functions**

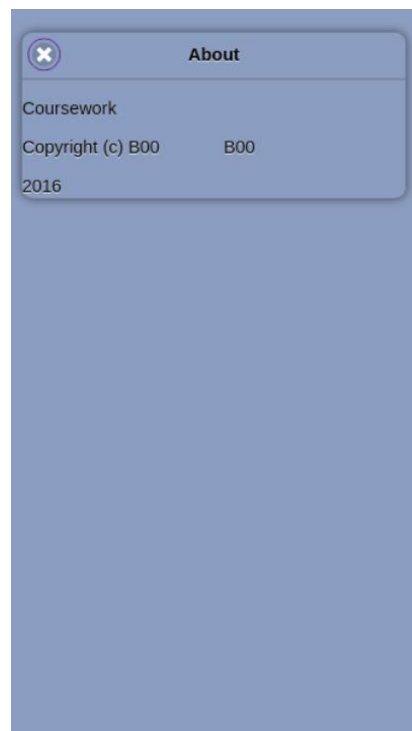
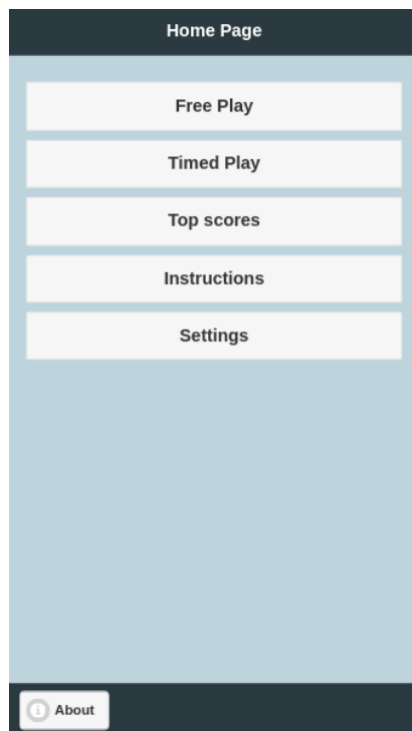
- *game\_loop\_free()*- *game\_loop\_timed()*: Those two functions controlled the game, they are launched whenever the user starts a new game. The animation is simulated by the *setTimeout()* call.
- *init()*: Resets the game state to start a new game
- *fill\_grid()*: Refills the grid with new/top boxes
- *draw()*: Draws the game in the canvas
- *mark\_alignments()*: Mark boxes to be removed using other functions to check the grid
- *removed\_marked()*: Remove marked boxes

### **Tests**

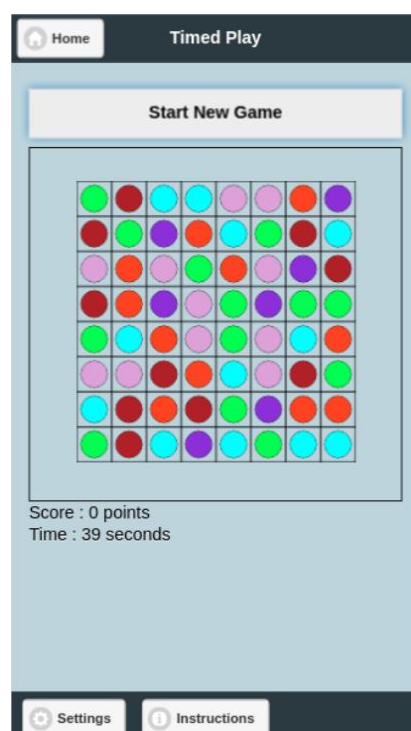
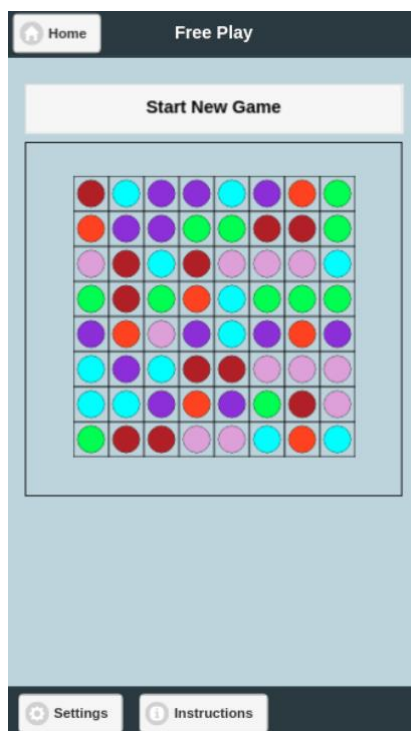
The project was tested using Webstorm simulated localhost sever and the Google Chrome debugger. We managed to correct every bug encountered.

## User Interface

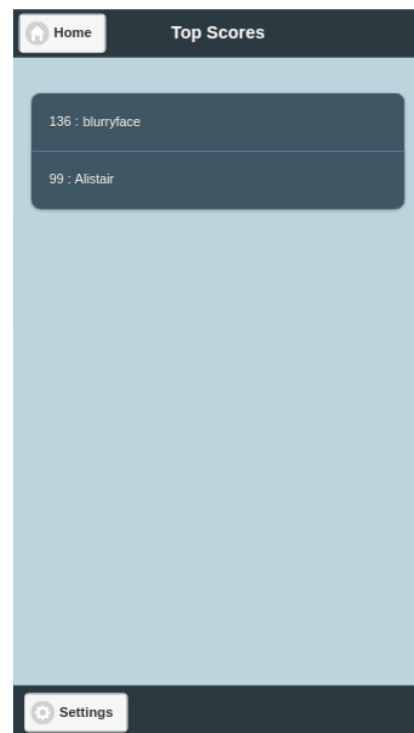
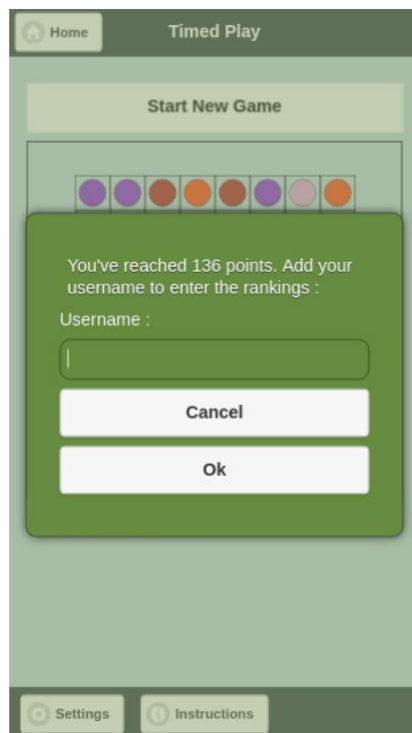
The UI design didn't change much since the first submission. On the homepage, we removed our banner IDs on the footer and added an *About* button that led to a page with our copyright information.



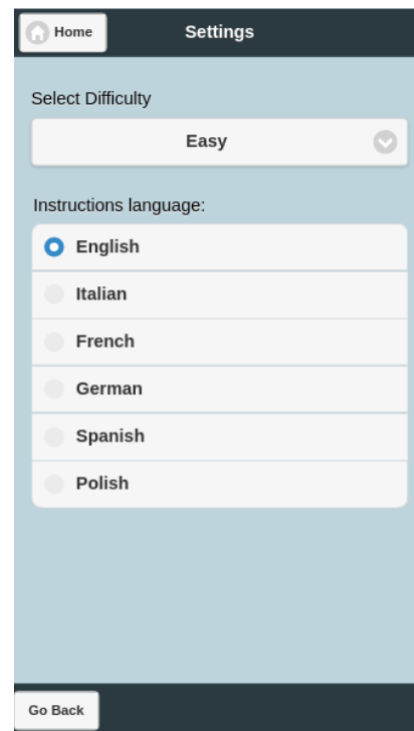
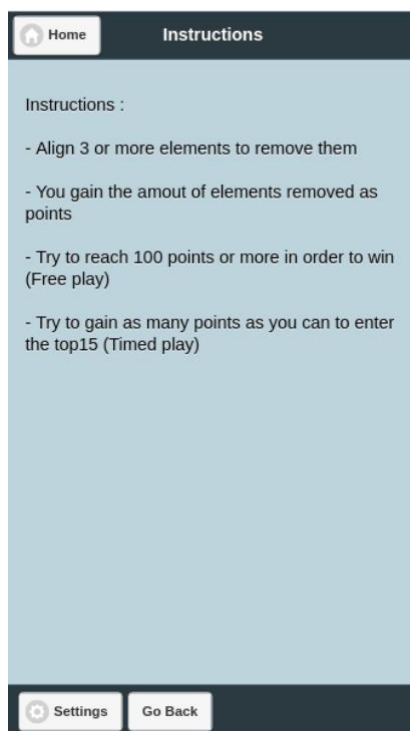
The two *Play* sections look similar, the only difference is the timer at the bottom of *Timed Play* which won't be the *Free Play* page. From the first submission, we added a *Start New Game* button.



After the timer in the *Timed Play* is over, the user will have to input a nickname, which will be stored in the *Top Scores* page with the number of points achieved.



We cleared the instructions for both types of Play mode and added more languages in the Settings.



### **Workload Allocation**

Having worked together on a project in the past, we knew we could trust each other regarding the project commitment. We split the work for actual coding (writing the functions or designing the user interface) however we kept the mapping of the game system, functionalities, and major UI designs for when we could actually talk face to face about it.

We had meetings every week, we would start by reviewing the code that we asked to write the week before (usually a couple of functions or pages) then we could discuss the system and design and attribute the following week workload, divided evenly amongst ourselves. The system worked well for us and we are both satisfied with how the project turned out.

### **Improvements**

If we had more time to develop the project, we would add more details. Firstly, when changing the language in Settings, the whole app would be translated and not only the instructions. We would have added more details on the *Top Score* page: it would actually show the top score for each difficulty level and not a general one.