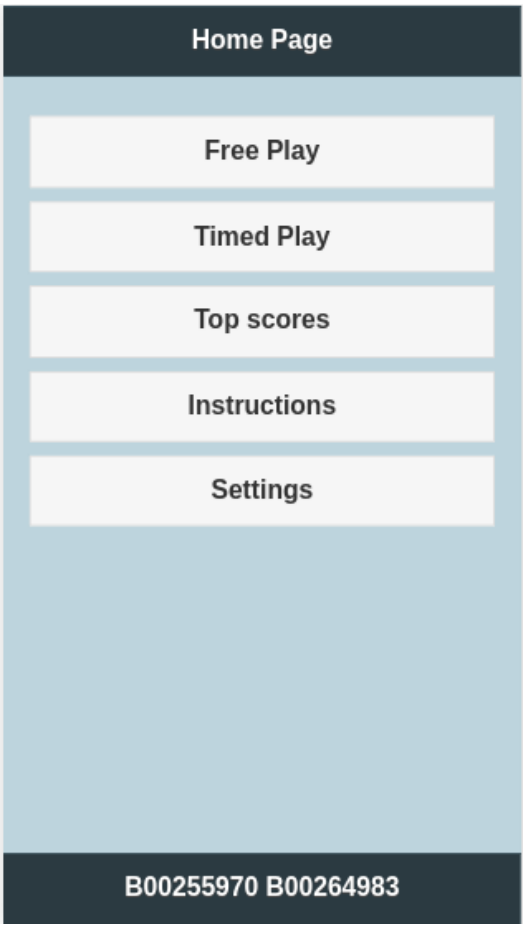
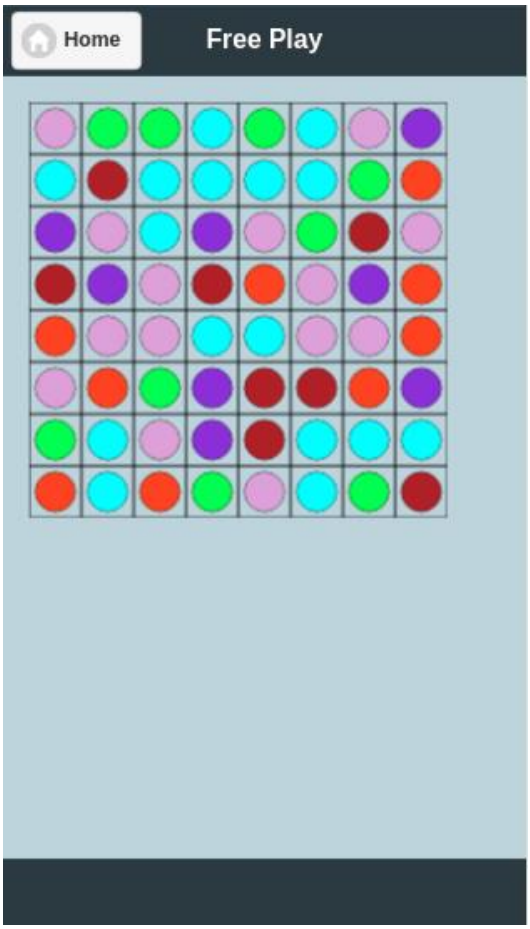
Assessment01: Project Specification

This project we submit will be a simple mobile game based off candy crush, which will have the following characteristics:

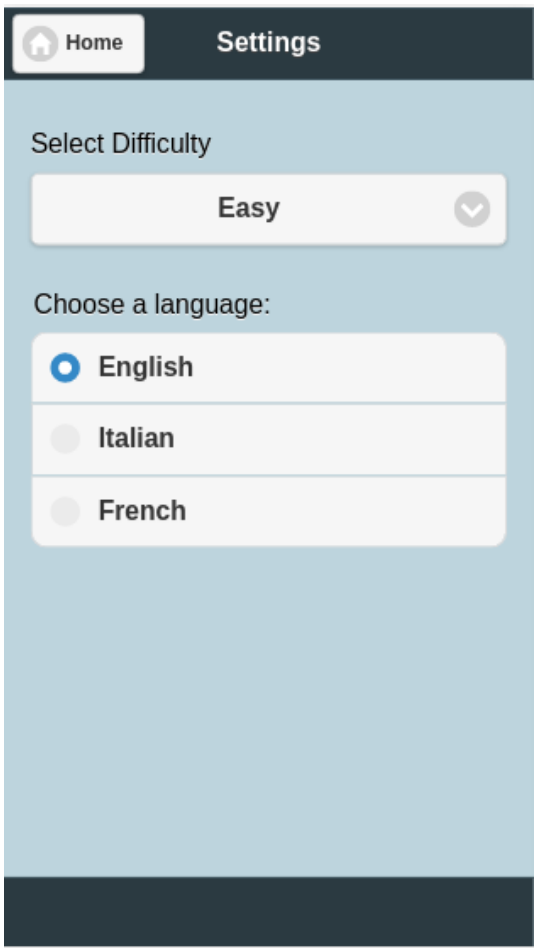
1. Home Page:
   The home page will consist of the game title, a logo designed by us and buttons leading the user to the other pages: the Game Area Canvas, instructions, top scores, settings and the option to play the game with a timer.
2. Game Area:
   The game area will be drawn on the canvas and updated through JavaScript according to user input.
3. Instructions:
   A simple page consisting of text explaining how to play the game, what the settings do and how the timer will interact with the gameplay.
4. Top Score:
   A normal game will end after 100 points and it is not recorded. Top scores will be included at the end of the game if "Timer" is selected.
   At the end of each timed game, the user can enter a name and the name and score will be saved in local storage and viewed in this page.
5. Settings:
   a. Level of game:
      The difficulty of the game can be adjusted. This is achieved by adding more colours that need to be matched together.
   b. Languages:
      The language of the game can be changed from English to French, Italian, German, Spanish and Polish.
6. Timer:
   Choosing this button will add a timer to the game and the score will be recorded. The difficulty of this game-mode can be adjusted in settings.

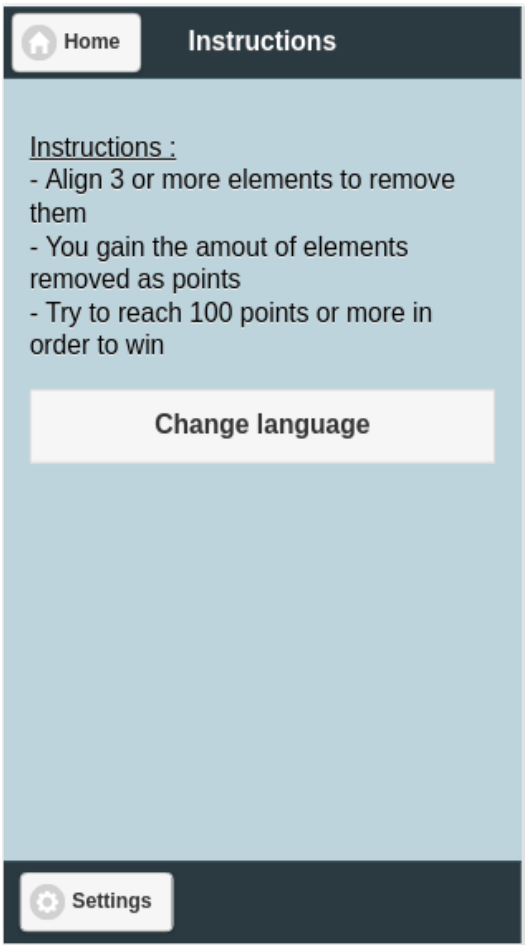
7. Operations through the graphical user-interface:
   a. Buttons:
      There will be 5 different types of buttons in this application:
      Buttons leading to the instruction page, top score page, settings page which you can adjust the difficulty and language, choosing the timer game-mode and a menu button.
   b. Gameplay:
      Gameplay will be achieved by dragging and dropping the appropriate circles to the desired square.





User-Interface Prototype





Design Documentation

Variables:

- Types : array that holds the possible colours
- Grid : two dimensional array that holds the whole game grid information
- Score : holds the score (how many blocks were matched)

Functions:

- is_horizontal_alignment: checks for horizontal alignments in the grid, returns true if an alignment is found.
- is_vertical_aligment: checks for vertical alignments in the grid, returns true if an alignment is found.
- swap : swaps two elements of the grid
- is_valid_swap: checks if the swap is valid using is_vertical_alignment and is_horizontal_alignment.
- mark_alignments: Goes through the grid and, if elements are matched, set the marked variable of those elements to true.
- remove_marked: removes marked elements from the grid
- Fill_grid: brings the elements down and creates new elements at the top randomly
- draw : draws the game on the canvas using all the grid's data
- game_loop: the main game loop with a settimeout to simulate the animations and the updating of data.
- get_mouse: retrieves the position of the mouse on the canvas
- convert_mouse_coords: converts the mouse coordinates from pixels to index elements for the grid.

Project Aims

We expect to implement the basic features of the game in the final application. This means that we can play the game and the game canvas will be updated accordingly to user input. The game will end after the player reaches 100 points, and it will be possible to reach the instruction page and change the language of the application.

We expect to be able to add another difficulty level to the game, a timer game-mode which also means that we will be adding the top scores and local storage.