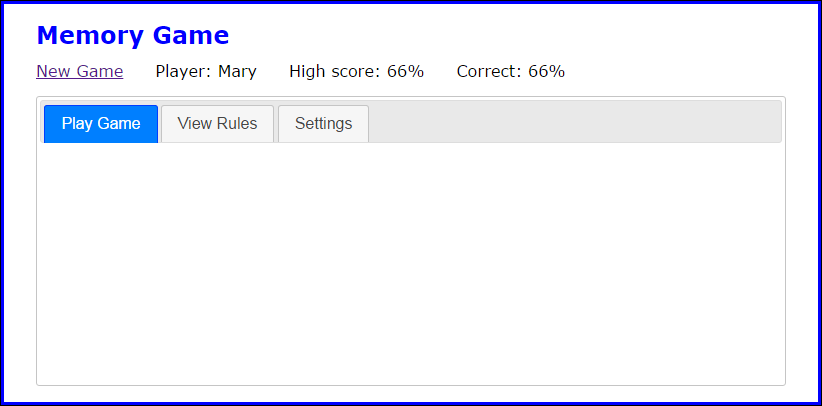
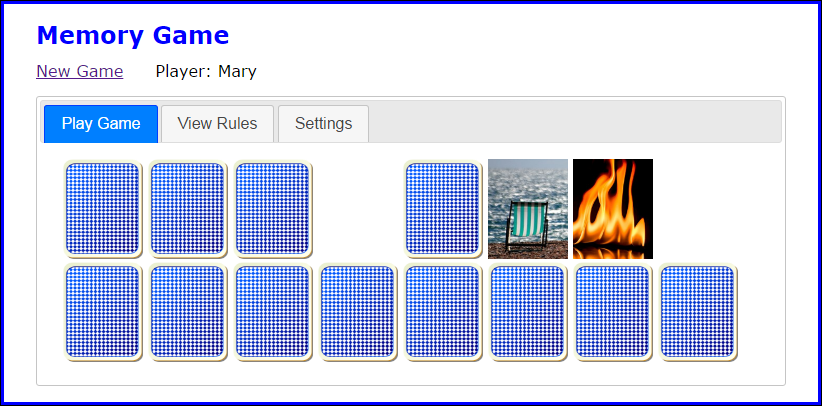
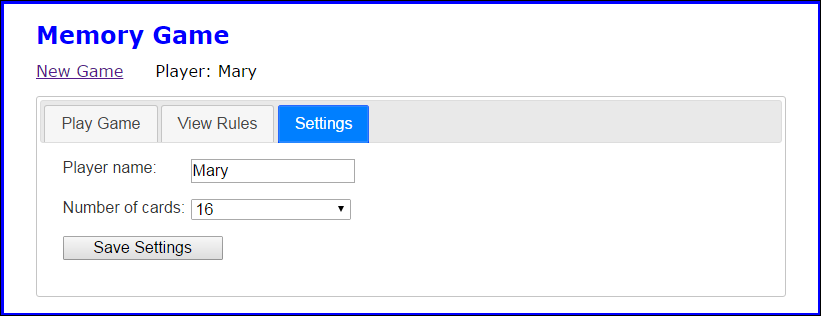
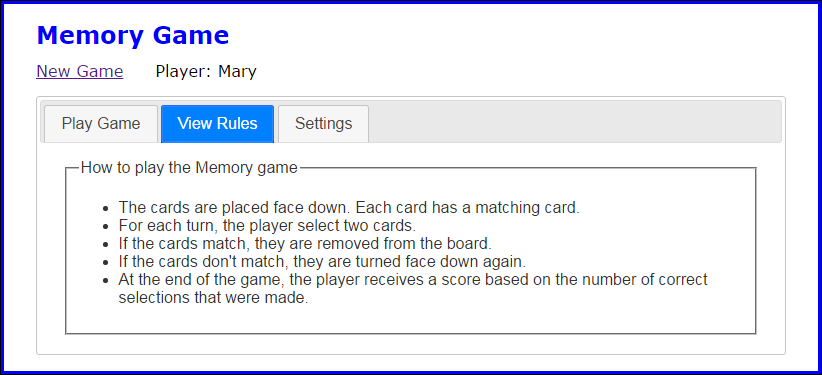
# Project: Create the Memory game

**Purpose:** For Project, you’ll create an application that allows the user to play the Memory game. The objective of this game is to turn over two cards at a time, trying to find cards that match until all cards have been matched.

**Sample User interface**



## Specifications

* The main portion of the user interface should be displayed in a Tabs widget with the three tabs shown above.
* Images are provided for 24 cards, the back of the cards, and a blank card. All these images should be preloaded when the application starts.
* By default, the game should display six rows of cards with eight cards in each row, for a total of 48 cards (two of each card). The user can change the number of cards used as well as the player name from the Settings tab.
* When the user clicks the Save Settings button, the player name and number of cards should be saved in session storage. In addition, the page should be reloaded so the player’s name, the player’s high score if previous games have been played, and the correct number of cards are displayed.
* Use an images array to store the images for the cards used by a game. Use a cards array to store the src attributes of the images for the cards that will be displayed on the board (two for each image).
* Create the HTML for the displayed cards by randomly selecting elements from the cards array. The cards for each row should be stored in a div element, and the div elements for all the rows should be stored in the div element with an id of “cards”.
* When the user clicks on a card whose back is displayed, the back of the card should be faded out over half a second and the front of the card should be faded in over half a second.
* When the user finds two matching cards, the cards should be hidden after one second using a sliding motion over half a second. If the cards don’t match, they should be faded out over half a second after two seconds and the back of the cards should be faded in over half a second.
* Each time the user completes a game, the user’s high score should be updated and displayed and the percentage of correct selections for the game that was just completed should be calculated and displayed. The high score should also be stored in local storage so it can be compared against the score for the user’s next game.
* When the game ends, the user can click the New Game link to start another game.

## The HTML for each card

* The HTML for each card should consist of an <a> element with its id set to the src attribute for the card and its href attribute set to “#”. The <a> element should contain an img element with its src attribute set to the source of the image for the back of the card and its alt attribute set to “”.

**Please refer to the Project Rubric for a complete breakdown of the grading scheme.**