Lab Activity:
Topic: The Document Object Model

1 Introduction

In this lab, you will modify an existing web memory game (version 1) based what you have learned in class so far, creating what I call here Version 2 of the game. As you probably know, the memory game is a simple game where the cards have images and are upside down on a table. Then the player flips two cards. If the figures match in both cards, these cards are removed from the table. In our case, we will not remove them, put place a blank image on them to simulate such a thing. If the figures don't match, the cards are flipped back and stay on the table.

Note: I am providing the version 1 of this code completely functional, but if you are a person that likes challenge like me, you could try to develop version 1 from scratch yourself. You will probably learn more by doing so, but it is entirely up to you.

2 The source code and images

The source code and images for version 1 of the memory game can be found at my github:

https://github.com/Paulo-Brasko/CS-1520/tree/master/labs/dom_lab

Study well the code being provided to you. Try to understand what each line does for the game. The more you understand it, the easier will be for you to convert this code into version 2.

3 Required code changes for Version 2

3.1 Create a JavaScript File

The first thing to be done in the development of Version 2, is to move all the JavaScript commands from the HTML file into a separate JS file, let's call it `Memory_Game_Version_2.js`. Create this file and link it in your HTML file, and run your code. It should reproduce the same behavior as the code in Version 1.

3.2 The creating of the image elements

In the Memory_Game_Version_1.html file, you will see about 24 repetitive statements used to create image tags:

```
<div id="wrapper">
    <div>
        <image src="/images/back_side.png" cardNumber="0"
        onclick=check_selection(event.target)></image>
    </div>
    <div>
        <image src="/images/back_side.png" cardNumber="1"
        onclick=check_selection(event.target)></image>
    </div>

    ...
</div>
```

Imagine if you want to display 300 images! You would need to type three hundred similar blocks! A waste of time and error prone! So, for your version 2 of the game, you want to create those statements programmatically using Javascript and the DOM object. The HTML file for version 2 should have only this:

```
<div id="wrapper"> </div>
```

Using the "`wrapper`" id to find this `<div>` tag in the document, create JavaScript commands to place all the game images (don't forget to surround the images with a second `<div>` as shown in the sample code in the previous page.)

3.2.1 Some hints

a) To create a new element, such as `div` or `img`, use `document.createElement();`
b) To set an image source in the `<img>` element, use `imageElement.src = "an image here"`. Remember at the beginning all images will be the back_side.png found in the `images` directory.
c) To set an event listener, use `imageElement.addEventListener("click", (e)=>check_selection(e.target))`
d) To set an attribute to the `img` tag: `imageElement.setAttribute("cellNumber", counter)` where `counter` is a variable going from 0 to 23
e) To add an element to another as a child: use `appendChild()` method
f) You need to repeat this procedure 24 times, so use a `FOR` loop to accomplish that

Create the above code inside a function, let's say `setupGame()`.

3.3 Implementing the number of trials

At the bottom of the web page, there is a field for placing the number of trials the player has currently performed. Create a JavaScript code that updates that field.

4 Extra credit

If you have time to spare and wiliness to take one more challenge, create a code logic that when all the cards are gone, a button is created with a caption "Play Again". If the player clicks on that button the whole game starts again with cards in new random locations.