

Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des pages Web »

**TP HTML5 – CSS3 - FlexBox - BootStrap
JavaScript - JQuery**

Apprentissage

Mise en pratique

Evaluation



TABLE DES MATIÈRES

1.	PROJET MEMORY POKÉMON (HTML5 - CSS3 - JQUERY).....	3
----	--	---

1. PROJET MEMORY POKÉMON (HTML5 - CSS3 - JQUERY)

Vous allez devoir développer un petit jeu de **Memory** basé sur les personnages **Pokémon**. Utilisation des langages **HTML5**, **CSS3**, **JavaScript/JQuery** (pas de Canvas). Les ressources graphiques sont jointes dans l'archive liée à ce TP. (200 cartes **Pokémon** et une carte back)

Petit rappel sur le jeu du **Memory** : Le jeu en phase initiale présentera un damier de cartes faces cachées (**4 lignes et 4 colonnes**).

Le but du jeu c'est de trouver **une carte** et **son double** dans le damier. À chaque fois que l'utilisateur clique sur le dos d'une carte, la carte se retourne en exposant un Pokémon, l'utilisateur doit cliquer une nouvelle fois sur une autre carte du damier, si la deuxième carte une fois révélée correspond bien à la première carte exposée, ce couple de carte disparaît à l'affichage dès lors que l'utilisateur clique sur une troisième carte, sinon les deux cartes retrouvent leur état initial (faces cachées).

L'utilisateur continue la partie, en essayant de mémoriser la position des cartes cliquées pour constituer les couples.

Le but du jeu c'est de parvenir à retrouver tous les couples de cartes en un minimum de cliques. Voici quelques captures d'écran pour vous aider à mieux comprendre l'objectif de ce TP.

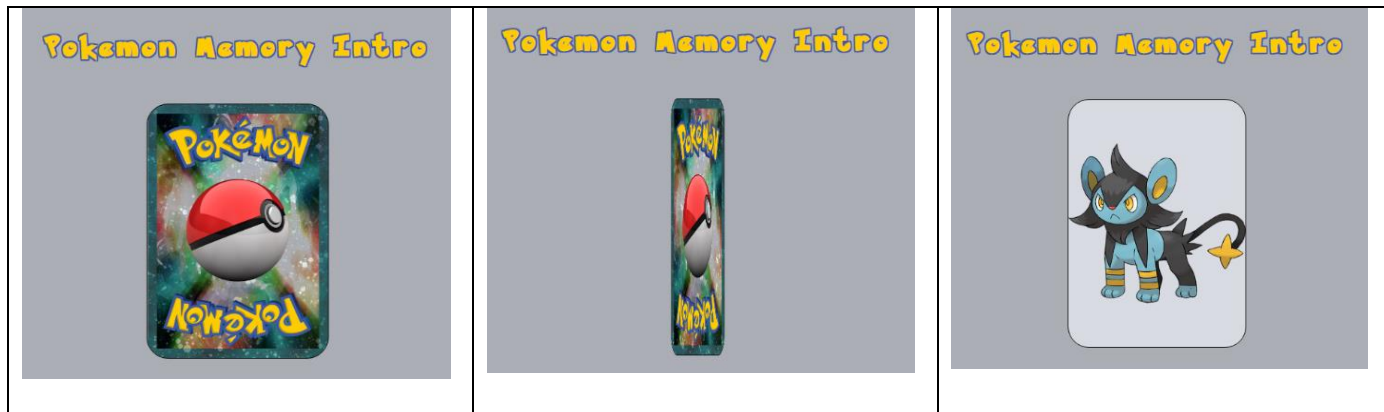


Afin de vous aider dans cette réalisation je vous propose de suivre ce cheminement :

Étape 1 :

Pour commencer, vous allez devoir implémenter une solution qui affiche une carte face cachée, lorsque l'utilisateur cliquera sur le dos de cette carte, elle effectuera une rotation de 90 degrés suivant l'axe des Y avant d'exposer son côté face. Vous pourrez utiliser le **plugin JQuery Transit** afin d'utiliser facilement les **transitions CSS3**.

<https://www.megaptery.com/2012/04/jquery-transit-un-plugin-pour-utiliser-facilement-les-transitions-css3.html>



Étape 2 :

Vous allez devoir implémenter une fonction JavaScript qui renvoie un **nombre aléatoire** compris entre les bornes **incluses min** et **max**. Les entiers **min** et **max** sont fournies en arguments à la fonction. Signature de la fonction :

```
function getRandomIntInclusive(min, max) { ... }
```

Étape 3 :

Vous allez devoir implémenter une solution qui créer un **Tableau** de **8 entiers aléatoires sans doublon**.

Étape 4 :

Vous allez devoir implémenter une solution qui **mélange** les éléments d'un **Tableau**. Pour cela, vous écrierez une **fonction** qui prend en **argument** un **Tableau** et qui retourne un **Tableau mélangé** et ceci de façon **aléatoire**.

Étape 5 :

En utilisant le travail effectué lors des [Étapes 1 à 4](#), vous allez maintenant créer un **Tableau** de 8 **Objets Littéraux** [{nom : « Pokemon » + numberaleatoire, img : « img/Pokemon » + numberaleatoire + « .png »}]

Puis vous **concaténerez** ce **Tableau** de 8 **objets** avec **lui-même** avant de le **mélanger**, vous obtiendrez ainsi un **Tableau de 16 objets** mélangés au sein duquel figure **8 paires différentes de cartes Pokémon** ;-)

Vous avez jusqu'ici tous les éléments pour pouvoir générer votre damier de **8 paires de cartes différentes** et pour débiter l'algorithme de déroulement du jeu ... Vous êtes libre de construire votre programme comme bon vous semble en utilisant vos connaissances acquises en **CSS3** et **JQuery**.

- Concernant le chiffrage du **score**, on parlera ici plutôt de « **handicap** » puisque dès lors que le joueur clique sur une carte il renforce son handicap, puisqu'il augmente son nombre de cliques. Le meilleur joueur sera celui qui aura le **handicap minimum**.

Vous comptabiliserez un **handicap supplémentaire** à **chaque nouveau clique** sur une carte **sauf lorsqu'une paire sera retournée**.

- Vous empêcherez tout nouveau clique sur une carte déjà retournée (qui expose donc un Pokémon).

La fin d'une partie exposera la dernière paire de Pokémon restante.



Étape 6 : Partie Développement pour Mobile

Fort de vos compétences acquises en développement pour Mobile et notamment avec l'utilisation du **Framework Apache Cordova**, vous devrez réaliser la **version mobile** de ce jeu de **Memory**. (Les captures d'écran suivantes vous aideront pour le maquetage de l'interface utilisateur, **damier 4 lignes par 4 colonnes**).

Pour ce faire, vous préparerez un nouveau projet nommé **Pokemon_Memory** avec la **CLI Cordova**, vous cibleriez le système **Android**.

Afin de rendre votre application totalement **Responsive** et si vous ne l'avez pas déjà fait, vous pourrez utiliser le **Framework Bootstrap** en restant sur du **Mobile First**.

Vous rajouterez un bouton **GO** de rechargement de la page afin de permettre à l'utilisateur de **relancer une partie**.

Enfin, vous utiliserez de façon spécifique les **images** contenues dans le répertoire **img**, elles ont été retravaillé pour s'adapter au développement Mobile. Pensez à tester cette application sur votre Smartphone !!!

