Memory

Marlière Jean-François

13 mars 2020

1 Présentation

Voici un petit jeu développé en PHP. Il est basé sur le principe du jeu de mémoire, où l'on a des paires de cartes cachées et il faut retrouver les paires une par une, parmi toutes les cartes.

Pour jouer, il faut cliquer sur une première carte, puis cliquer sur une deuxième. Si elles sont identiques, la paire est retrouvée et les cartes restes affichées, sinon elles sont à nouveau cachées. Tant que toutes les paires de carte ne sont pas retrouvées, la partie continue.

Un compteur de temps comptabilise le temps en secondes que dure la partie.

Une barre de progression affiche votre progression.

2 Description du code

Ce jeu est développé en PHP, avec jQuery, SASS et un peu de SQL.

La grille est générée avec un numéro de partie, les cartes, leurs positions aléatoires, puis sauvegardée en base de donnée, pour ensuite être affichée. Cela se fait côté PHP, la suite est gérée en javascript et jQuery.

Lorsque l'on clique sur une première carte, elle est affichée et mémorisée. Lorsque l'on clique sur une deuxième carte, elle est aussi affichée et mémorisée. Ensuite les deux cartes sont comparées :

Si elles ne sont pas identiques, au bout d'une seconde on les cache à nouveau, et on peut à nouveau choisir une autre première carte, et un ainsi de suite...

Si elle sont identiques, on les laisse affichées et on sauvegarde en base de donnée que la paire est retrouvée, en basculant la propriété *perdue* de 1 à 0, de chacune des cartes. Une fonction Ajax est utilisée.

2.1 Fichiers PHP

2.1.1 Classes

— *Grille* : permet de générer la grille avec les cartes.
— *Carte* : représente une carte présente dans la grille.
— *memoryBDD* : contient les méthodes pour faire les requêtes SQL.

2.1.2 Autres fichiers

— *index.php* : fichier de départ de notre jeu, où tout s'éxécute.
— *ajax.php* : permet d'exécuter une requête SQL via jQuery avec AJAX.

2.2 CSS

— saas/memory.sass : Contient le code qui permet de générer le fichier de style CSS

Une boucle permet de générer automatiquement les classes permettant d'afficher l'image des cartes à partir du sprite *images/cards.png*.

On utilise *background-position-x* et *background-position-y* pour sélectionner l'image contenue dans le sprite.

2.3 Javascript

— js/memory.js : Contient tout le code java-script et jQuery traitant les événements d'affichage du jeu

3 Travail à faire

3.1 Tester le jeu et terminer au moins une partie. Analysez son fonctionnement

3.2 Etudier le code en vous aidant ses commentaires

3.3 Modifiez le Jeu

Après avoir étudié le comportement du code, vous êtes en mesure de comprendre son fonctionnement.

3.3.1 Texte de la barre de progression

Lorsque l'on a terminé la partie, la barre de progression doit afficher 100% et être totalement pleine.

Hélas, afficher un message final avec la fontion *alert()* provoque un bug. La mise à jour du DOM est bloquée tant que l'alerte est à l'écran. Il faudrai que la barre soit correctement mise à jour avant que le popup n'apparaisse.

Trouvez une solution à ce bug, en n'utilisant pas la fonction *alert()* mais une fenêtre modale par exemple.

3.3.2 Charger une grille existante

Dans la classe *Grille*, il y a la méthode *chargerGrille()* qui permet de charger la grille que l'on vient tout juste de générer. Il serait intéressant de pouvoir charger une grille déjà existante en base de donnée, pour reprendre une partie si elle a été interrompue :

— Ajoutez au dessus de la grille un champ texte et un bouton, qui permettra de saisir un numéro de partie.

— Modifiez le code en partant de la méthode chargerGrille() de la classe Grille pour permettre le chargement d'une grille existante en base.

3.3.3 Une partie plus lognue

Actuellement nous avons 14 cartes différentes. Notre sprite *cards.png* en contient 18. Modifiez le code pour que la grille contienne 18 paires au lieu de 14, sur un affichage de 6 lignes et 6 colonnes.

3.3.4 BONUS : Refactoring des données

Nous avons en base de données 2 tables : grille et carte. Pour des raison de performance, nous souhaitons n'avoir qu'une seule table avec les même informations.

Modifier la base de donnée et le code pour qu'une seule table soit nécessaire pour enregistrer la grille avec les cartes.

3.3.5 SUPER-BONUS : Animation des cartes

Le client du jeu est exigeant et souhaiterai que lorsque l'on clique sur une carte, une rotation de la carte (sur l'axe vertical)est réalisée pour afficher l'autre face, comme si on retournait une vrai carte.

Modifier le fichier SASS pour permettre cette transition.

Petit indice : Il est possible d'utiliser cette propriété CSS :

```
transform: rotateY(90deg)
```