

Développement d'Applications Réticulaires

Travaux sur machines, 7 novembre 2016
Pratique de JavaScript, DOM et gestion d'événements

1 Présentation

Cet exercice a pour but de vous familiariser avec le langage JavaScript, l'API de manipulation du document et la gestion d'événements du navigateur.

Pour ceci, vous allez devoir développer un petit jeu classique de travail de mémoire pour enfants en utilisant les primitives du navigateur, sans l'aide de bibliothèques.

Il s'agit d'un jeu de cartes contenant des paires facilement identifiables. Ici chaque carte aura un fond coloré sur lequel est imprimé une lettre de l'alphabet dans une autre couleur. Au départ, on dispose toutes les cartes en grille sur la table, face cachée. Puis, le jeu se déroule ainsi :

1. on retourne une carte,
2. on en retourne une seconde,
3. si les deux cartes forment une paire, on les laisse ainsi, sinon on les remet face cachée,
4. et on recommence jusqu'à avoir tout retourné.

Un modèle à compléter vous est donné :

- http://benjamin.canou.fr/Cours/DAR_2016/P1/game.html
- http://benjamin.canou.fr/Cours/DAR_2016/P1/game.js

La solution vous est donnée :

- http://benjamin.canou.fr/Cours/DAR_2016/P1/solution.html
- http://benjamin.canou.fr/Cours/DAR_2016/P1/solution.js

2 Création d'éléments DOM

Dans un premier temps, écrivez une fonction JavaScript de rendu du plateau de jeu.

Le plateau de jeu initial vous est donné sous forme de deux tableaux :

- `cards` est un tableau donnant pour chaque paire de cartes sa couleur de fond, sa couleur de texte et sa lettre.
- `state` est un tableau à deux dimensions d'objets indiquant en chaque emplacement du plateau de jeu la carte (champ `card` contenant un indice dans le tableau `cards`) et si la carte a déjà été retournée (champ `flipped` initialement à `false`, puisque toutes les cartes sont disposées face cachée au début du jeu).

Vous pouvez vous inspirer de la structure du document généré par la solution (en l'observant avec F12), mais ce n'est en rien obligatoire, tant que le rendu est similaire. De même, des classes CSS vous sont fournies en option. Les cartes doivent être de simples élément DOM (div) dont le style définit les couleurs. L'utilisation d'images ou de canvas est hors sujet.

À noter qu'il existe deux possibilités pour la suite : soit vous générez une fois pour toute la structure de la page et vous effectuez des modifications locales en place, soit vous refaites le rendu complet à chaque changement. Ce choix est à décider dès maintenant puisque votre fonction de rendu doit donc être écrite en conséquence. La solution fait le choix d'un seul rendu initial et de modifications locales au cours de la partie.

3 Gestion d'événements

Implémentez la boucle de jeu, sur le modèle de la solution (vous pouvez bien sûr faire mieux). Le code doit résister aux clics intempestifs (durant le retour visuel, une fois le jeu terminé, etc.).

- le joueur clique sur une carte,
- s'il clique ensuite sur la même carte ou sur une carte déjà retournée, on annule le premier clic,
- s'il clique sur une carte pas encore retournée, on la retourne temporairement, et
 - si la carte fait bien la paire, on affiche un témoin vert pendant un moment et on passe au tour de boucle suivant en mettant à jour le compteur de paires restantes,
 - sinon on affiche un témoin rouge un bref instant, permettant au joueur de faire travailler sa mémoire, puis on remet les cartes face cachée,

Pour finir, on affiche un message de victoire.