

Puissance 4

Pour vous rafraîchir la mémoire, je vous ai trouvé des exercices corrigés qui illustrent l'utilisation de JQuery / Ajax / ...

Moodle sur JQuery

[Exercices TP-2](#)

Utilisation avancé de jquery, animation notamment

[Exercices TP-3](#)

Utilisation d'Ajax

[Exercices TP-5](#)

On va essayer comme dans leur exemple de créer un jeu de puissance 4, un peu comme dans l'exemple du morpion qui est présenté dans le document.

Partie 1 : Réaliser un puissance 4 en version client uniquement

Voici l'ordre dans lequel vous devrez procéder pour faire ce travail :

- 1) Créez la page html, et créez votre puissance 4 sous forme d'un tableau (rappel, le puissance 4 dispose de 7 colonnes sur 6 lignes).
Vous devriez pour vous aider donner des id différents à chaque ligne et à chaque colonne, par exemple (case_2_4) pour la case ligne 2 colonne 4.
en en-tête vous pouvez mettre les informations du jeux (joueur 1 / joueur 2) avec qui est en train de jouer
- 2) Les jetons des joueurs sont rouges et jaunes, vous pouvez par exemple colorer la case en utilisant une classe (css) spécifique pour le faire.
- 3) Ecrivez une fonction d'initialisation en javascript/jquery qui remet la partie à zero
- 4) N'oubliez pas de lancer le jeux en lançant la fonction d'initialisation en utilisant le document-ready de JQuery
- 5) Programmez un événement sur le survol du tableau et mettez un indicateur sur l'en-tête de colonne pour connaître la colonne en cours de survol qui sera prise en compte par le clique (vous pouvez gérer ça avec les coordonnées de la souris par rapport à la taille des colonnes par exemple)
exemple : td de 200px, je suis en 1345px
donc $\text{ceil}(1345/200) = 6$ donc je serais sur la 6^é colonnes (en partant de 0)

- 6) Programmez les événements de click sur les td du tableau puissance 4, pour détecter un clique sur une case.

Lors du clique sur cette case, il faudra faire des contrôles et des interactions :

- Est-ce que la colonne sur laquelle on clique est pleine (si oui on ne peut pas jouer cette colonne)
- Faire une mini animation (en JQuery avec animate) qui représente la descente du jeton

A la fin de l'animation, on effectue les contrôles suivants :

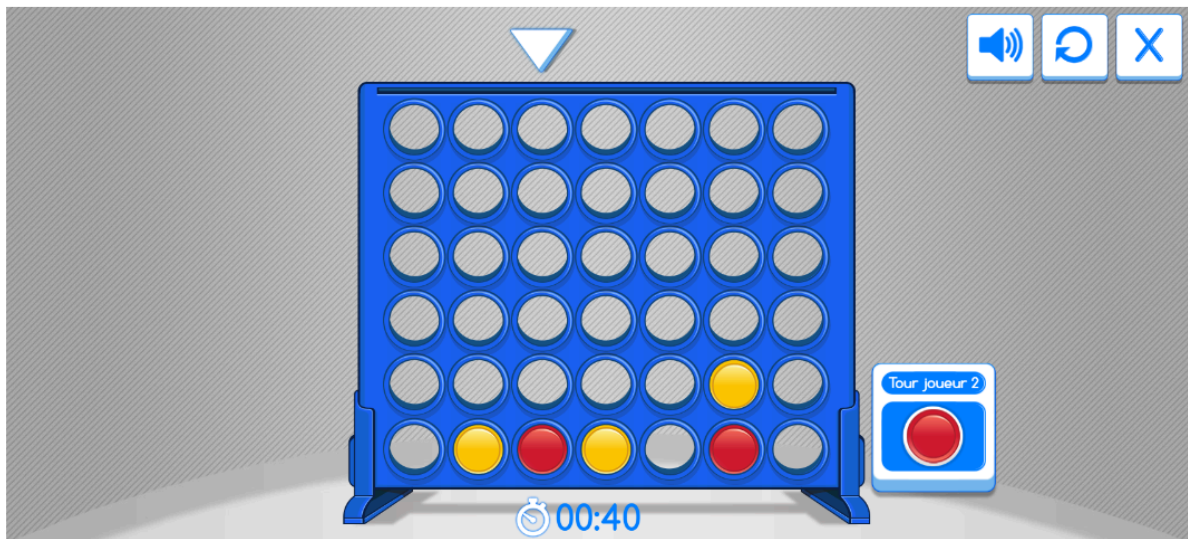
- Est-ce que j'ai fait un puissance 4 ?
- Est-ce que le tableau est plein ? (match nul)

- 7) Lors de la fin de la partie sur victoire / nul : on affiche un message à l'utilisateur pour signaler la victoire. On modifie les informations du joueur pour incrémenter son nombre de victoire, on incrémente le nombre de parties jouées. Puis on lui propose de relancer une nouvelle partie, qui pourra lancer la fonction reset (la même qu'au début)

Partie 2 : Ajouter des "interactions" serveurs

- 1) Créer une base de données (simple) qui contiendra les tables suivantes :
 - a) partie_en_cours
 - b) scores
- 2) A la fin de la partie, vous mettrez à jour la table des scores avec une requête ajax (\$ajax, \$.get ou \$.post)
- 3) Maintenant on va essayer de sauvegarder la partie en temps réel, chaque fois qu'un joueur joue un coup, envoyer une requête ajax, contenant l'état de la partie.
- 4) Lorsqu'on charge la page, si une partie est en cours, recharger la avec les données qui sont contenues en base de données

Voici un exemple (photo non contractuel) :



Vous n'êtes pas obligé d'en faire autant graphiquement.

Algorithme de validation/victoire

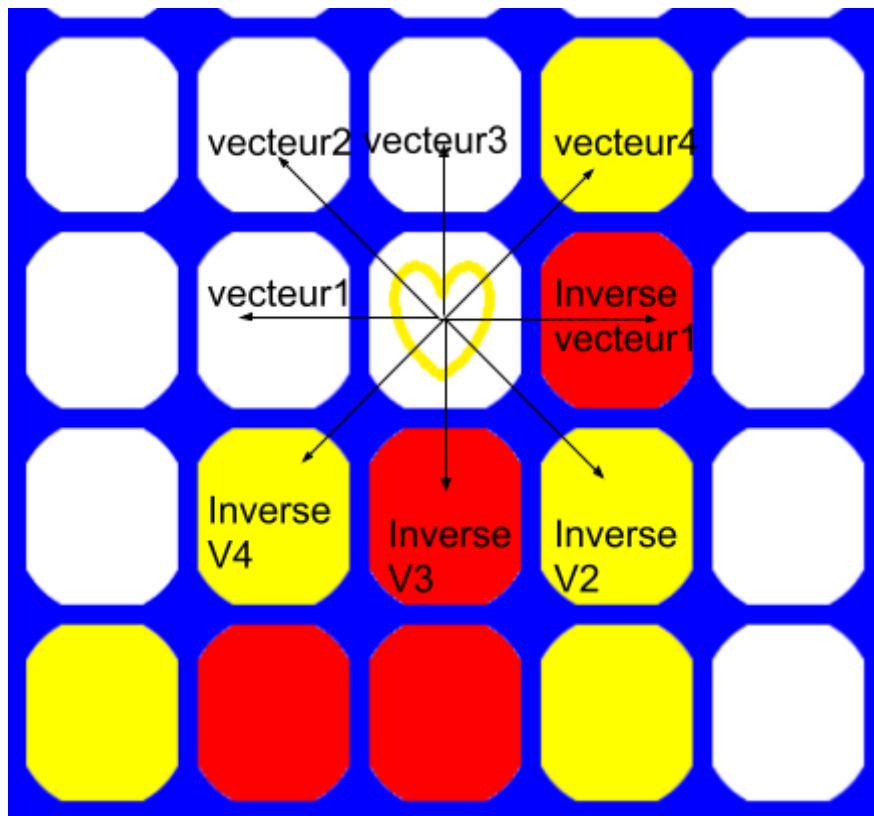
```
FONCTION validation(IN caseJouer, IN Couleur)
Vecteur1 <- [0,-1]
Vecteur2 <- [1,-1]
Vecteur3 <- [1, 0]
Vecteur4 <- [1, 1]

nbJeton <- validerVecteur(caseJouer+Vecteur1, Vecteur1, Couleur, 1)
nbJeton <- validerVecteur(caseJouer+InverseVecteur(Vecteur1), InverseVecteur(Vecteur1), Couleur, nbJeton)
SI nbJeton >= 4
Victoire(Couleur)
FIN SI

....
....
FIN FONCTION

FONCTION validerVecteur(IN case, IN Vecteur, IN Couleur, IN nbJeton)
SI case+vecteur est de couleur = Couleur
RENOYER validerVecteur(Case+Vecteur, Vecteur, Couleur, NbJeton +1)
FIN SI
RENOYER nbJeton
```

A répéter pour chacun des vecteurs (voir l'infographie suivante)



Dans l'exemple :

validerVecteur pour le Vecteur 1 renverra : 1
 validerVecteur pour l'inverse du Vecteur 1 renverra 1
 Donc sur l'horizontal, on a 1 seul jeton (celui qu'on vient de poser)

validerVecteur pour le Vecteur 2 renverra : 1
 validerVecteur pour l'inverse du Vecteur 2 renverra 2
 Donc sur le \ on a 2 jetons (celui qu'on vient de poser + 1)

validerVecteur pour le Vecteur 3 renverra : 1
 validerVecteur pour l'inverse du Vecteur 3 renverra 1
 Donc sur la vertical, on a 1 seul jeton (celui qu'on vient de poser)

validerVecteur pour le Vecteur 4 renverra : 2
 validerVecteur pour l'inverse du Vecteur 4 renverra 4 (2+2)
 Donc sur le / on a 4 Jeton, c'est gagné