

Evaluation JS - jQuery (notée sur 20 points, avec 6 points bonus)



Le but de l'exercice est de reproduire la mécanique du jeu Duck Hunt en multijoueur, avec un joueur qui contrôle le canard et l'autre qui doit lui tirer dessus.

Pour votre inspiration : <https://www.youtube.com/watch?v=-1NylsZXeqU>

À la fin du projet, vous devrez archiver (.zip) votre projet de jeu et l'envoyer à l'adresse webmaster@washaweb.com.

L'archive ZIP devra être nommée ainsi: **prénom-nom-eval-duckhunt.zip**, *exemple : georges-dupont-eval-duckhunt.zip*

1 - Gestion des déplacements (2 points)

Créer un div carré représentant le canard. Le carré doit pouvoir se déplacer dans tout la fenêtre.

AIDE (1)

1. créer une div carré bleu (100px) en position absolute.
2. quand on presse BAS, changer la propriété top, pour déplacer le carré vers le bas
3. Faire la même chose pour DROITE, HAUT, GAUCHE

Ajouter une écoute du clavier '**keydown**' et trouver le **event.keycode** correspondant.
Changer la propriété CSS **top** avec la variable **"+=1%"**

2 - Attention aux collisions (1 points)

Le cube peut ne peut se déplacer que dans une espace restreint : Une aire de jeux. Quand il atteint le bas, il se téléporte en haut Donc si la position est inférieur à 0%, la position devient égale à 100%.

AIDE (2)

Il va falloir ajuster notre traitement:

1. Définir une variable "posTop" qui va stocker cette position verticale.
2. Quand la touche pressée est BAS, on incrémente la variable "posTop".
3. Si la limite est atteinte, on ajuste la valeur.
4. On effectue l'animation.
5. Faire la même chose pour une variable "posLeft" pour le déplacement horizontal.

3 - Ajout du joueur à la souris (2 points)

Lorsque la souris clique sur le carré :

- Le carré devient rouge.
- L'écoute du clavier s'arrête.
- Déclencher une popup : le jeu est terminé, victoire du joueur à la souris !

AIDE (3)

Utiliser les méthodes jQuery `.on()` et `.off()` pour ajouter /supprimer un événement sur le carré.

4 - Cas de victoire pour le joueur au clavier (2 points)

Nous allons autoriser le joueur au clavier à gagner s'il arrive à échapper au joueur souris pendant plus de 30 secondes. Ajoutez un timer au jeu qui fera gagner le joueur au clavier.

- Au bout de 30 secondes, la fonction `victoireClavier` se lance
- Le cube devient vert
- Les event listener sont désactivés
- Déclencher une popup : le jeu est terminé, victoire du joueur au clavier!

AIDE (4)

- Utiliser une fonction `setTimeout(nomDeFonction, 30000)`.
- Utiliser un gestionnaire `.off()` pour supprimer l'événement du carré.

5 - Score (2 points)

Les deux joueurs ont un score et dispose de 5 ou 8 manches.

AIDE

1. Demander à chaque joueur son pseudo (nom).
2. Créer deux éléments HTML pour afficher les pseudos et les score.
3. Créer une variable pour stocker le score de chacun des joueurs.
4. Augmenter cette valeur à chacune de leur victoire et mettre à jour la zone HTML correspondante.
5. Si le score dépasse 5, un message félicite le gagnant.

6 - Personnalisation (1 points)

- Changer le curseur pour une cible par exemple.
- Changer le joueur au clavier avec l'image (animée ?) d'un canard par exemple.
- Changer le background de la scène, améliorer l'écran des scores, les zones d'affichage des noms des joueurs...

7 - Équilibrage automatique (3 points)

Si le joueur à la souris gagne trop souvent, on augmente la vitesse de déplacement du carré et diminue sa taille, inversement si le joueur au clavier gagne trop souvent, on diminue la vitesse de déplacement du carré et augmente sa taille.

8 - Modo solo : CPU challenge (3 points)

- Au début du jeu, ajouter un écran de choix du mode de jeu : solo ou multijoueur.
- Si il n'y a qu'un seul joueur, créer un nouveau mode solo qui utilise uniquement la souris pour le joueur, le carré se déplace alors de façon aléatoire dans votre page (comme dans le jeu original).

9 - Modo solo avancé (4 points)

- Pour le mode solo avancé, ajouter plus d'une cible, augmenter le nombre de cibles au fur et à mesure des niveaux de jeu.
- Mode multi avancé : Il est aussi possible de jouer avec un joueur à la souris, un autre au clavier et 1 CPU...

Rappels : Pour obtenir un chiffre aléatoire :

```
function getRandomInt(max) {  
    return Math.floor(Math.random() * Math.floor(max));  
}  
//récupère un chiffre aléatoire entre 0 et 100  
var randomNumber = getRandomInt(100);
```

BONUS 1 - Ammo (2 points)

Gérer des munitions : l'utilisateur à la souris dispose d'un certain nombre d'essai pour réussir. Ce nombre est affiché à l'écran sous forme de munitions.

BONUS 2 - CPU Challenge (2 points)

Gérer l'augmentation de la difficulté à chaque niveau dans une fonction du type :

```
function gameStart(nbCibles, cibleSpeed, ammo){  
    //traitement...  
    return true si le niveau est fini en gagnant,  
    return false si le niveau est perdu (ammo==0)  
}
```

BONUS 3 - Make it awesome challenge (*2 points*)

- Gérer les trois états de la cible : en vol, shootée, morte. Adapter les comportements et sprites à chacun des états.
- Gestion d'un *cheatcode* pour avoir les munitions en illimités en mode solo.
- Gestion d'un *cheatcode* pour ralentir le mouvement des canards en mode solo.