

# M3206

## Présentation et objectif

Durant cet exercice, nous allons aborder les techniques de drag@drop avec JQuery.

L'objectif étant de créer un jeu type "puzzle" puis "point and click".

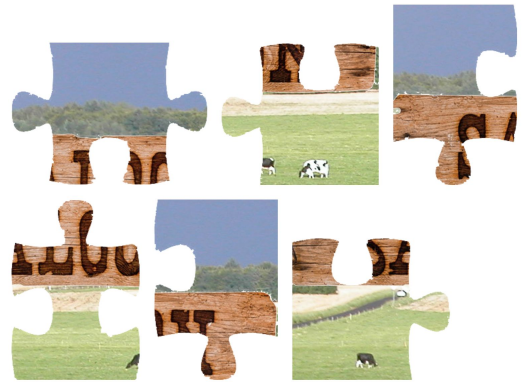
ce type d'interaction pourra vous servir lors de votre projet S3

## Réalisation d'un mini puzzle avec CSS+jquery

### 1. Mise en place des pièces du puzzle

- Placer les 6 morceaux de puzzle les uns à côté des autres (dans le désordre bien sûr).
- Pour cela, utilisez simplement des `<img>`

(lien 6 pieces:



[http://149.91.88.82/fabrice/lienszip/images\\_puzzle.zip](http://149.91.88.82/fabrice/lienszip/images_puzzle.zip))

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="utf-8">
  <style type="text/css">
    ...
  </style>
</head>
<body>
  <img ... piece1.png... >
  <img ... piece2.png... >
  ...
</body>
</html>
```

### 2. Gérer l'animation du drag&drop avec jquery

- Ajouter jquery + jquery-ui à votre page html (voir la doc en ligne)

- Tapez le code jquery nécessaire pour que les pièces soient déplaçables (voir la documentation de jquery -> draggable ). On ne gèrera pas ici la vérification des positions finales des pièces.

resultat: <https://youtu.be/O85ckCZ3UUU>

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta charset="utf-8">
  <style type="text/css">
    ...
  </style>
</head>
<body>

  <img ...piece1.png.....>
  <img ...piece2.png..... >
  ....

</body>

<script src="....jquery...."></script>
<script src="jquery-ui....."></script>

<script>
  .....draggable.....
</script>
</html>
```

ici nous gérons le déplacement des pieces mais aucune vérification de leur bonne position finale lors du drop

## Gestion d'un "point & click"

### Point and Click

trouver le parchemin du docteur Knok  
petit indice: il était sur le point de le détruire...



tester le résultat: <http://149.91.88.82/fabrice/exo2/exo02.html>

pour ne pas perdre de temps, quelques indices:

- on cherche moins bien dans le noir
- détruire un parchemin par le feu est efficace
- une clef trouve toujours sa serrure...

### 3. Gestion d'un click sur une zone de l'image

Créez une "scène" (un div) de 1280x853px avec une image de background quelconque (position:relative avec marge auto).

Créez dans cette scène (ce div) une img (une image de bouton) en position absolute. Mettre un id "monbouton" sur cette image. ajouter la propriété CSS "cursor: grab;" pour que l'image de la souris change lors du survol.

Créez une autre img dans la "scène" avec une image quelconque (mettre cette img en position absolute). Lui donner un id "macible".

Créez un événement `$("#monbouton").click(function(){ ... })`

Dans cet événement faire disparaître l'img "macible" avec la méthode `$("#macible").toggle();`

La méthode `toggle()` fait disparaître ou apparaître à chaque fois l'élément.

Il est aussi possible de d'utiliser la méthode `.hide()` ou `.show()` ou `.fadeIn(500)` ou `fadeOut(500)`.

## 4. Gestion du son

Déclarer un élément audio avec la balise `<audio>` (voir la doc de la balise audio)

```
<audio id="son" loop preload> <source src="ambiance2.mp3" type="audio/mpeg"> </audio>
```

Attention, depuis 2018 il n'est plus possible de démarrer un son au démarrage de la page (pour ne pas polluer de façon intrusive la navigation des internautes).

Il faut passer par un bouton par exemple pour lancer le son.

```
document.getElementById("son").play();
```

ou raccourci: `son.play();`

(voir aussi `son.pause() ...`)

Avec ce principe, ajouter un son qui se jouera sur le click du bouton précédent.

## 5. Gestion d'un drag&drop d'un objet avec vérification de ses coordonnées de dépôt.

Sur le même principe que le `drag@drop` vu dans l'exercice du puzzle, placez un élément (une img en absolute) dans le div de la scène.

Rendez cette image déplaçable ( `xxxx.draggable( )` )

Ajoutez l'événement généré lors du lâché de l'objet :

```
$("#monObjetQuiBouge").draggable( { stop: function(e, ui){ console.log(ui.position.top);  
console.log(ui.position.left) } } );
```

A la place des `console.log()`, il est ensuite possible de tester un lâché sur une zone particulière de la scène en faisant par exemple :

```
if(ui.position.top>340 && ui.position.top<380 && ui.position.left>845 && ui.position.left<880 ) {  
  
    console.log("déposé là où je le voulais"); }
```

## Gestion d'un mini jeu de "point@click"

### Point and Click

trouver le parchemin du docteur Knok  
petit indice: il était sur le point de le détruire...



Réaliser un mini jeu point@click en vous basant sur celui ci-dessus

<http://149.91.88.82/fabrice/exo2/exo02.html>

pour ne pas perdre de temps, quelques indices:

- 1) on cherche moins bien dans le noir
- 2) détruire un parchemin par le feu est efficace
- 3) une clef trouve toujours sa serrure...

---

## Trop facile pour vous ?

Si vous avez terminé, vous pouvez maintenant faire une scène plus complexe avec plusieurs objets à déplacer, plusieurs boutons à actionner, même mettre des gifs animés pour animer la scène (bougie qui vacille, horloge qui tourne,...).