

Développement Web Avancé

L3

Coverflow

07/01/2019

Réalisé par

Mahmoud HAMDAN

Sommaire :

- **Instructions**
- **Mon fichier JavaScrip**
- **Mon fichier CSS**
- **Mon fichier HTML**

• Instructions

Tout d'abord j'ai réussi à créer un Coverflow uniquement en CSS et HTML (que je vous ai déjà montré), en suite il n'était pas possible de ne pas créer mes fonctions dans un fichier JS pour travailler sur le fonctionnement des flèches -> <- directionnelles (droite & gauche), ainsi que les clicks next et Prev pour mettre les opérations en ordre plus proprement, pour les configurations de Plugin même si mon projet CoverFlow doit contenir des opérations en CSS plus qu'en JS, j'ai fait beaucoup de recherche sur internet concernant mon projet CoverFlow, j'ai décidé donc au final d'utiliser des bibliothèques JavaScript dont j'avais besoin, j'ai donc utilisé :

- La bibliothèque Modernizr : Qui permet de communiquer avec le navigateur et développée en JavaScript détectant l'implémentation native par le navigateur de différentes fonctionnalités récentes provenant des spécifications HTML5 et CSS3.
- La bibliothèque jQuery : Pour permettre de faciliter les fonctionnalités communes de Javascript et bien pour permettre de gagner du temps de développement de mon projet CoverFlow, ceci possède donc les fonctionnalités suivantes :
 1. Manipulation du DOM en HTML ou CSS.
 2. Gestion des événements (clic, survol, soumettre un formulaire).
 3. Effet d'animation.
 4. Plugins.

• Mon fichier JavaScript

CoverFlow principale :

J'ai donc mon image principale et j'ai des images derrières par la fonction responsiveGallery et la fonction getTransitionCSS qui est lié avec le fichier CSS pour contrôler l'animation de vitesse lors de la modification des propriétés CSS.

Je met des petites flèches à droite et à gauche pour choisir la direction des mes images depuis la fonction MouveCoverFlow par ces deux opérations :

```
opts.$btn_next.on('click',function(e){
    !anima && MouveCoverFlow(+1);
});
opts.$btn_prev.on('click',function(e){
    !anima && MouveCoverFlow(-1);
```

```
});
```

Où il y a +1 pour l'image suivante et -1 pour l'image précédente.

Les condition suivantes me présentent les images en ordre par la transformation CSS :

```
if(compteur === 1){ //c'est l'image présente numéro 1
    formCSS = getTransform1CSS();
} else if(compteur === 3){ //l'image présente numéro 3
    formCSS = getTransform3CSS();
} else if(compteur === 5){ //l'image présente numéro 5
    formCSS = getTransform5CSS();
```

J'ai donc mon image principale et j'ai des images derrières par la fonction responsiveGallery et la fonction getTransitionCSS qui est lié avec le fichier CSS pour contrôler l'animation de vitesse lors de la modification des propriétés CSS.

CoverFlow Horizontale :

```
var rotateY = 0;
```

Là je définis une rotation pour déplacer les élément autour de l'axe des ordonnées, sans le déformer.

```
var rotate_int = 0; je reviens ver ma rotation initiale.
```

```
var ry = 360/size; c'est pour que l'utilisateur quand il clique une fois sur l'image pour la faire
pivoter à 360 degrés et que la page de lien s'affiche.
```

```
var box = 0; et enfin il vide le rotatY, il est réutilisable.
```

Ainsi que je gère les directions des clicks (next + prev) comme le Coverflow principal mais là j'ai la fonction mouvement_slid où elles se trouvent les opérations :

```
$(".next").on("click", function(){
    rotate_int -=1;
    mouvement_slid();
});
```

```
$(".prev").on("click", function(){
    rotate_int +=1;
    mouvement_slid();
});
```

CoverFlow verticale :

elle est pareil que *CoverFlow Horizontale* sauf que l'opération "transform":"**rotateX**("+z+"deg)
translateZ(« +translateZ+"px)" me permet à gère le type du rond / coverflow pour qu'elle soit

verticale par rotateX et pour effectuer une rotation à un élément HTML par rapport à l'axe de perspective X.

• Mon fichier CSS

Mon fichier CSS, J'ai indiqué la rotation, la profondeur et de quel point il sera prolongé, comme dans #carousel j'ai dit que je veux le largeur de mon coverflow horizontal qu'il soit 100%; ainsi que pour le #carousel2 c'est bien pour mon coverflow vertical, j'ai dedans l'effet -webkit-transform-style: preserve-3d; qui formate les éléments en utilisant des transformations 3D.

J'ai aussi dans #carousel figure la propriété opacity qui spécifie l'opacité ou la transparence d'un élément.

Je donne donc une place ou je déplace les 3 coverflow (horizontal, vertical et le principal) par :

```
.Coverflowhorizontal {  
width: 100px;  
margin: 0 auto;  
}
```

ET

```
.CoverflowVertical {  
width: 1100px;  
margin: 0 auto;  
}
```

Et bien pour le coverflow principal :

```
position: absolute;  
left: 0;  
top: 0;  
z-index: 0;
```

coverflow : { rotate: 50, stretch: 0, depth: 100, modifier: 1).

J'ai créé le background de mon site, j'ai donné mes instructions, la taille en largeur et en longueur, j'ai passé au CSS transforme + le 3D.

• Mon fichier HTML

Mon fichier normal ou j'ai appelé les fichiers / dossier comme :

```
css/coverflow_style.css  
lib/modernizr.custom.js  
jquery-1.11.2.min.js  
coverflow3options.js
```

Et ainsi que la classe responsiveGallery qui se trouve dans les elle est lié avec le fichier coverflow_style.css pour l'affichage des images en coverflow.

```
$(function () {  
    $('.responsiveGallery-wrapper').responsiveGallery({  
        animatDuration: 400,  
        $btn_prev: $('.responsiveGallery-btn_prev'),  
        $btn_next: $('.responsiveGallery-btn_next'),  
    });  
    $('#carousel').covhoriz3D();  
    $('#carousel2').covrtical3D();  
});
```

Dans cette fonction j'ai donc la vitesse de mouvement de mes images pour le coverflow principal qui est animatDuration: 300,