

2016

Rapport : Cat Dress

Rapport du projet en web : jeu d'habillage, dans les langages
HTML/CSS/JavaScript

Nafissa Van-Elsen
26/05/2016



Présentation du projet

Pour le projet en web, j'ai fait le choix de concevoir un jeu, qui s'intitule « Cat Dress ». Le but étant de faire habiller un avatar, ici représenté par un chat, avec un accessoire de tête, une coupe de cheveux, et un vêtement, se trouvant à côté de l'avatar.

Dans ce jeu, il y a la possibilité de l'habiller en « Fille » ou en « Garçon ». J'ai également mis un bouton « Retirer » permettant de tout recommencer.

Les langages utilisés sont le HTML, CSS, mais principalement le JavaScript, à l'aide du Framework JQuery, qui contient la grande majorité du code du jeu.

Pour la réussite du projet, il a fallu plusieurs critères:

- La possibilité de déplacer les objets, et de les faire magnétiser sur l'avatar, et au bon endroit
- Mettre qu'un seul élément de chaque catégorie (un accessoire, une coupe de cheveux, et un vêtement) sur l'avatar
- Les faire ramener à leur emplacement d'origine si on les retire ou s'ils ne sont pas mis sur l'avatar
- De faire changer le genre de l'avatar, grâce aux gestionnaires d'évènements qui seront liés aux boutons
- Et de pouvoir retirer les objets, grâce à un autre gestionnaire d'évènement lié au bouton « Retirer »

Ce projet se constitue de trois fichiers : « accueil.html », « style.css », et « script.js ».

Fonctions du jeu

Pour commencer le code, il fallait qu'on sache comment la fenêtre en HTML sera organisée : j'ai donc fait deux tableaux dans le fichier html, dont l'un représente les objets, l'autre tableau l'avatar.

En effet, grâce à ces tableaux, il sera possible de positionner les éléments comment il le faut, c'est-à-dire, on a bien la tête, et le corps du chat qui sont bien positionnés et alignés l'un en dessous de l'autre. Pour les objets, la même chose, j'ai aligné les accessoires de tête sur la même ligne, les coupes de cheveux, ainsi que les vêtements sur leur ligne correspondante. Ainsi, l'affichage est clair, on sait où il faut piocher les éléments.



Les première et deuxième lignes du tableau « objet », correspondent respectivement aux accessoires, et à la coupe de cheveux qui peuvent se mettre sur la tête (1) du chat. La dernière ligne est composée des vêtements, donc ils vont sur le corps du chat (2). L'utilisateur a la possibilité de mettre qu'un élément pour chaque catégorie d'objets, c'est-à-dire qu'on peut mettre qu'un accessoire, qu'une coupe et qu'un vêtement. Et pour ce faire, il faut passer par le codage en JavaScript que j'expliquerai par la suite.

J'avais dit précédemment que l'on peut gérer le genre de l'avatar, j'ai donc implémenté deux boutons permettant de faire ce changement. Ainsi, lorsqu'on appuie sur le bouton garçon ou fille, les objets changent, ainsi que le fond du jeu.

J'ai également mis la possibilité de tout recommencer, les objets reviennent à leur place respective.



Fonctions JQuery

```
$('.deplacer').mousedown(function(event)
```

Les éléments doivent être mis sur l'avatar, en cliquant dessus et en restant appuyé sur un objet. Les objets déplaçables ont la classe « deplacer » et ils seront donc déplacés seulement si l'utilisateur va appuyer avec la souris sur l'objet.

```
$('.en-mouvement').removeClass('en-mouvement');  
boite.addClass('en-mouvement');
```

Lorsque l'utilisateur aura appuyé sur la boîte (donc un élément), l'objet va avoir la classe « en-mouvement », jusqu'à qu'il le relâche.

```
$('html').mousemove(function(event)
```

Ensuite, on suit les mouvements de la souris sur toute la page grâce au sélecteur jQuery \$('html'), grâce au gestionnaire d'évènement « mousemove ». Cette fonction permet également à ce que la souris ne perde pas la boîte si on la ferait sortir de l'objet « en-mouvement ».

```
$('html').mouseup(function(e)
```

Après, dès qu'on relâche l'objet, on a une très grande fonction sur l'évènement « mouseup », qui m'a posé quelques difficultés, puisqu'il faut gérer l'emplacement de la boîte par rapport à l'avatar et de remettre la boîte à sa place si elle est mal placée.

```
var x = boite.offset().left+50;  
var y = boite.offset().top+50;  
var posX= $("#tetechat").offset().left;  
var posY= $("#tetechat").offset().top;  
var large=$("#tetechat").width()+$("#tetechat").offset().left;  
var haut=$("#tetechat").offset().top +$("#tetechat").height() +$("#bustechat").height()+$("#piedchat").height();
```

Les variables x et y représentent la position de la boîte sur l'axe x et y par rapport à la page. J'ai rajouté 50 pixels aux coordonnées de la boîte, ainsi on vérifie que l'objet, et non pas seulement que le coin de l'objet, est dans la zone de l'avatar qui lui correspond.

Les variables posX et posY correspondent aux coordonnées du coin de la case « #tetechat », et les variables large, et haut, sont les largeurs et hauteur du corps de l'avatar.

```
if (x>posX && x<large && y>posY && y<haut)
```

Grâce à ces variables, on peut en déduire des conditions pour le placement de la boîte qui dit que : «si la position x de la boîte se trouve entre le coin de la tête de l'avatar (posX) et la largeur de l'avatar, et la position y de la boîte se trouve entre le coin de la tête de l'avatar et sa hauteur, alors la boîte va obtenir la classe « poser », et bien sur cette boîte se positionnera sur la case correspondante.

Exemple pour la tête :

```
if (boite.hasClass('accessoire'))
{
    for (var i=0;i<3;i++)
    {
        if(tableau1[i].hasClass('poser'))
        {
            tableau1[i].removeClass('poser');
            tableau1[i].offset(tableau2[i].offset());
        }
    }
    tableau1[0].css('z-index','100');
    tableau1[1].css('z-index','100');
    $('.en-mouvement').offset($('.a').offset());
    $('.en-mouvement').addClass('poser');
    $('.en-mouvement').removeClass('en-mouvement');
```

Tout d'abord j'ai fait deux tableaux (tableau1 et tableau2) qui permettent de : pouvoir obtenir les cases qui seront les objets à déplacer (ils ont la classe « caseN° »), et leur emplacement d'origine permettant de les faire revenir la dessus si besoin (classes « placeN° »).

Ainsi si on veut déplacer un accessoire sur la tête, tout d'abord on va parcourir le tableau1 pour voir s'il n'y a pas d'objet déjà posé sur la tête, pour le remettre dans son emplacement initiale marqué dans le tableau2, et pour qu'on puisse remplacer par la nouvelle boîte. Cette nouvelle boîte va se positionner sur la tête de l'avatar grâce à la classe « .a », qui correspond au positionnement de l'accessoire sur l'avatar, et on remplace la classe « en-mouvement » par la classe « poser ».

```
else
{
    var classe=['case0','case1','case2','case3','case4','case5','case6','case7','case8'];
    for (var i=0;i<9;i++)
    {
        if(boite.hasClass(classe[i]))
        {
            boite.removeClass('en-mouvement');
            boite.offset(tableau2[i].offset());
        }
    }
}
```

Sinon, si l'objet ne se trouve pas au bon positionnement de la boîte par rapport à l'avatar, on va le remettre à sa case d'origine : on enlève donc la classe « en-mouvement », et on le remet dans sa place d'origine, grâce au parcours de la boucle for dans le tableau2.

Difficultés : Récupérer les positions de la souris et créer une zone permettant de magnétiser sur l'avatar, cela m'a pris beaucoup de temps, par la recherche des fonctions permettant d'englober tout le tableau du chat.

Remettre les objets à leur place, j'ai pris la décision de mettre de nouvelles classes pour faciliter le remplacement des objets en capturant la position de chaque case contenant la classe « place + n° »

```

$('#bouton').mousedown(function(e)
{
    for (var i=0;i<tableau1.length;i++)
    {
        if(tableau1[i].hasClass('poser'))
        {
            tableau1[i].offset(tableau2[i].offset());
        }
    }
});

```

Quand il y a des objets déjà placés on peut les retirer grâce à l'implémentation du bouton « Retirer » et grâce à la fonction qui gère l'évènement sur le bouton. En effet, lorsqu'on clique sur le bouton, on va parcourir le tableau1 qui va rechercher tous les objets contenant la classe « poser » et de les remettre à leur place toujours grâce au tableau2 qui contient les places correspondantes à chaque boîte sur le chat.

```

$('#Garcon').mousedown(function(e)
{
    for(var i=0;i<image1.length;i++)
    {
        image2[i].hide();
        image1[i].show();
    }
    $('#fond').css('background-image',' url("wall1.png")');
});
$('#Fille').mousedown(function(e)
{
    for(var i=0;i<image1.length;i++)
    {
        image1[i].hide();
        image2[i].show();
    }
    $('#fond').css('background-image',' url("wall2.png")');
});

```

On peut choisir le genre « fille » ou « garçon » du chat, et c'est par la gestionnaire d'évènement sur les boutons « fille » et « garçon ». Encore, il fallait deux tableaux, image1 qui contient les images des objets garçons, et image2 qui contient les images des objets filles. Ainsi si par exemple, on veut mettre l'avatar en fille, on cache les images « garçon » et on fait apparaître les images « fille ». Et pour une meilleure visibilité du jeu, je change en même temps le fond du jeu par le css() de la div id= « fond » qui englobe le jeu.