

1. POO EN JAVASCRIPT / JQUERY – LA COURSE DE VOITURES

L'objectif de ce TP est de vous permettre de vous familiariser avec la **Programmation Orientée Objet en JavaScript**. Pour cela, vous allez développer dans un premier temps une petite application très simple de **course de voitures** en respectant les spécifications **EcmaScript 5**. Jusqu'ici, vous avez déjà rencontré comment utiliser des **Objets Littéraux** dans un programme (par exemple pour le **jeu de la Bataille**). Ici, vous allez apprendre comment **créer** des **Objets** et comment les **utiliser** dans votre programme.

Vous allez devoir concevoir une course de voitures, pour cela vous devrez créer plusieurs **Objets Voitures** qui **partagent** les **mêmes propriétés** et **fonctionnalités**.

Pour cela, vous devrez utiliser une fonction spéciale de la **POO** que l'on appelle un **Constructeur**, en fait c'est une fonction qui permet de créer des **Objets** de *même nature*.

A chaque fois que vous appelez un **Constructeur**, celui-ci crée un **Objet** et lui attribue des **propriétés intégrées**.

Pour **appeler** un **Constructeur** il faut utiliser un mot-clé de **JavaScript** : **new**

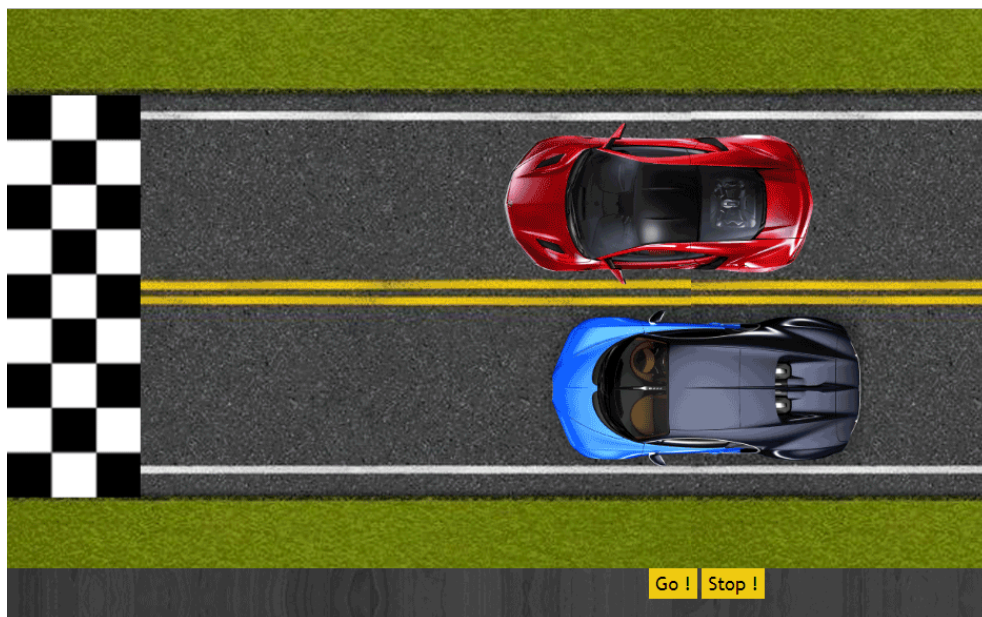
On doit l'utiliser comme il suit :

```
var voiture = new Voiture (param1, param2, ...);
```

Ici, on déclare utiliser le **Constructeur Voiture** auquel on passe les **paramètres** qu'il attend et celui-ci retourne un **Objet** que l'on assigne à une **variable voiture**.

En **JavaScript** et dans la plupart des langages de programmation objet, le nom du **Constructeur** **commence** par une **majuscule** pour pouvoir le distinguer des autres fonctions.

Voici le rendu de l'application que vous devrez réaliser :



Pour y arriver, vous devrez créer dans un premier temps un **Constructeur Voiture** qui vous permettra d'afficher à une **position donnée** de votre page web une **image** de voiture.

Vous aurez donc besoin de créer une fonction **Constructeur** que vous appellerez **Voiture** et qui sera dotée de propriétés de positionnement dans la page et qui fera référence à une **méthode** de **dessin** à laquelle vous passerez un **paramètre image**.

```
var Voiture = function (x,y, img) {  
    this.x = x;  
    this.y = y;  
    this.dessiner(img);  
};
```

Voilà à quoi devra ressembler votre **Constructeur**, noté l'utilisation du mot-clé **this** qui fait référence à l'**Objet** voiture que vous créerez lorsque vous ferez appel à ce **constructeur**. Voici les appels qui permettent de créer **2 Objets** à partir du **constructeur Voiture** que l'on assigne à 2 variables **car1** et **car2**.

```
var car1 = new Voiture(1000,105, "img/car1.png");  
var car2 = new Voiture(1000,280, "img/car2.png");
```

Dans un premier temps, vous devrez écrire la méthode **dessiner (img)** et comme cette méthode devra être associée au **constructeur Voiture** vous devrez utiliser la propriété **prototype** de **JavaScript**, elle permet d'attribuer facilement une **même fonctionnalité** (sous forme de méthodes) à **différents objets**.

En **JavaScript**, tous les **constructeurs** ont en effet une propriété **prototype** à laquelle vous pouvez **ajouter des méthodes**. Et toutes les méthodes que vous pouvez ajouter dans le **prototype** d'un **constructeur** seront ensuite disponibles dans l'**ensemble des objets** créés avec ce dernier. Voici comment écrire la méthode **dessiner()** en l'associant à la propriété **prototype** du **constructeur Voiture** :

```
Voiture.prototype.dessiner = function(img) {  
  
    // Placez votre code JavaScript ici  
  
};
```

Ainsi, après l'écriture de cette méthode et après la création de vos « **objets** » **car1** et **car2** vos deux voitures seront positionnées sur votre page web ;-)

Maintenant, vous me direz que tout ceci manque cruellement de mouvement !!!

Vous allez devoir y remédier, pour cela vous devrez écrire une nouvelle méthode associée au **prototype** du **constructeur Voiture** qui permettra de mettre en mouvement les **objets** créés à partir de ce **constructeur**.

Vous appellerez cette méthode **deplacerAGauche(distance)**, cette méthode implémentera la décroissance de l'**abscisse x** de l'**objet** concerné. Pour le paramètre **distance** vous pourrez choisir un **nombre aléatoire** compris entre [0,15].

Pour obtenir **un effet de mouvement**, vous pourrez utiliser la fonction **setInterval()** de **JavaScript** qui sera déclenchée sur un clic du bouton **Go !** et qui permettra de relancer la méthode **deplacerAGauche()** toutes les **20 ms**.

Pour arrêter le mouvement, vous devrez implémenter le code nécessaire avec le gestionnaire d'évènement associé au clique sur le bouton **Stop !**

Vous devrez implémenter dans un deuxième temps une solution qui respecte les spécifications de **EcmaScript 6** avec l'écriture d'une classe et de ses méthodes.

Remarque : Si vous êtes plus à l'aise avec la Bibliothèque **JQuery**, rien ne vous empêche de l'utiliser ici ;-)

Maintenant que vos objets Voitures sont créés, vous pouvez envisager de créer une véritable course avec la mise en place d'un **compte à rebours** avant le démarrage des voitures et d'un gestionnaire d'évènements sur les touches clavier de votre choix. La frappe sur une touche clavier permettra au joueur de faire avancer sa voiture vers la gauche, il devra essayer d'être plus rapide que son adversaire. Dès qu'une voiture franchira la ligne d'arrivée, un message s'affichera pour féliciter le gagnant de la course.

Je vous conseille de créer un nouveau projet qui reprendra les éléments nécessaires à la construction de vos divers éléments.

Les captures d'écran suivantes vous permettront de mieux envisager le travail à réaliser ;-)

Les divers éléments graphiques qui composent l'interface de cette application vous seront fournis dans une archive.

