

# **Contrôle de Web Avancé / Javascript**

## **e-services FA - Vendredi 25 Octobre 2018**

---

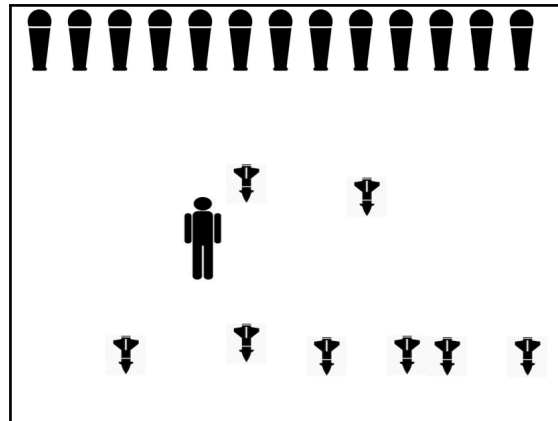
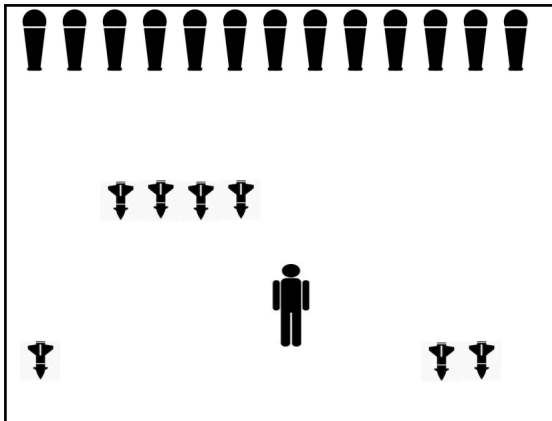
Nom :

Prénom :

---

Aucun document ni périphérique électronique n'est autorisé.  
Répondez aux questions sur les pages gauches laissées blanches.

Le contrôle porte sur un petit jeu où un « petit bonhomme » doit éviter des missiles qui tombent et doit parvenir à toucher au moins une des rampes.



L'idée n'est pas ici de coder entièrement le jeu mais juste le démarrage, c'est-à-dire :

- un petit bonhomme qui peut se déplacer en haut, en bas, à gauche et droite selon des touches du clavier
- d'afficher un cadre et des canons/rampes en haut de celui-ci.
- de faire tomber des missiles à intervalles non réguliers.

Il y a 1 objet *Jeu* et 4 classes :

- *ImageDeplacable* qui définit le principe de placer une image à un endroit donné (position absolue).
- *PetitBonhomme* qui hérite d'*ImageDeplacable*, qui affiche un petit bonhomme et qui permet de piloter celui-ci à l'aide de touches spécifiques
- *ObjetQuiTombe* qui hérite d'*ImageDeplacable* et qui fait parcourir l'image de haut en bas selon un rythme (codé en dur)
- *Canon* qui affiche un canon, et qui va tirer des missiles à intervalles irréguliers.

L'objet *Jeu* et la classe *ImageDeplacable* vous sont données ci-dessous.

```

jeu = {
  x : 10,
  y : 10,
  largeur : 400,
  hauteur : 300,
  espaceEntreRampe : 30,
  positionCanons : [],
  demarrer () {
    this.creerCadre();
    this.creerRampeMissile();
  },
  creerCadre () {
    this.cadre = document.createElement("div");
    var style=this.cadre.style;
    style.position="absolute";
    style.left=this.x+"px";
    style.top=this.y+"px";
    style.width=(this.x+this.largeur)+"px";
    style.height=(this.y+this.hauteur)+"px";
    style.borderColor="black";
    style.borderWidth="2px";
    style.borderStyle="solid";
    document.body.appendChild(this.cadre);
  },
  creerRampeMissile () {
    for (let dx=this.x+12 ; dx < this.x+this.largeur ;
        dx+=this.espaceEntreRampe ) {
      new Canon(dx,this.y+4,this.y+this.hauteur);
    }
  }
}

```

```

function ImageDeplacable (
    taille = {largeur: 30, hauteur: 30},
    urlImage = "forme.jpg",
    x = 200, y = 20
) {

  this.creerBalise(urlImage, x, y, taille.largeur, taille.hauteur);
}

ImageDeplacable.prototype.creerBalise =
  function (urlImage, x, y, largeur, hauteur) {
    this.balise = document.createElement("img");
    this.balise.src=urlImage;
    var style=this.balise.style;
    style.position="absolute";
    style.left=x+"px";
    style.top=y+"px";
    style.width=largeur+"px";
    style.height=hauteur+"px";
    document.body.appendChild(this.balise);
  }

```

Les fichiers image disponibles sont « missile.jpg », « petitBonhomme.jpg », « canon.jpg ».  
Le lancement de l'application est fait comme suit :



```
window.onload = function () {  
    jeu.demarrer();  
    bonhomme=new PetitBonhomme({});  
}
```

## Exercice n°1 : PetitBonhomme

Cette classe est à écrire à l'aide de prototype et non en mode Javascript 6.

### A) Définition de la classe et du constructor

Définissez la classe *PetitBonhomme* où le constructeur prend un objet comme seul paramètre. Cet objet doit avoir les propriétés *touches*, *taille*, *urlImage*, *x* et *y*.

Ces propriétés ont les valeurs par défaut suivantes :

- touches >> {gauche : 37, haut : 38, droite : 39, bas : 40}
- taille >> {largeur : 50, hauteur : 70}
- urlImage >> "petitBonhomme.jpg"
- x >> 200,
- y >> 200

Indiquez que cette classe hérite d'*ImageDeplacable* (toujours pas de Javascript 6).

Le constructeur appelle le constructeur de sa classe mère (avec les bons paramètres) puis invoque la méthode *fixerEvenementTouches* (voir plus bas).

### B) Getter/setter

Définissez un getter et un setter pour les propriétés *x* et *y* de toutes instances de la classe. Ces getter/setter utiliseront les propriétés de style *left* et *top* de la balise/image de l'instance.

### C) Gestion des touches

Implémenter la méthode *fixerEvenementTouches* de la classe *PetitBonhomme* qui prend en paramètre un objet comme suit {*gauche*, *haut*, *droite*, *bas*} où la valeur de chaque propriété est un entier correspondant au code clavier pour le haut, le bas...

Dans cette méthode, ajouter un auditeur à l'événement *keydown* au *body* de la page HTML. Comme il pourrait y avoir plusieurs petits bonhommes, et donc plusieurs instances (avec des codes claviers différents), l'événement *keydown* doit pouvoir avoir plusieurs auditeurs.

Cet auditeur doit déplacer le petit bonhomme selon les touches appuyées.

Les précédents getter/setter seront ici bien utiles.

Méthode utile : *parseInt*.

Propriété utile : *keyCode* d'un événement.



## Exercice n°2 : ObjetQuiTombe

Cette classe est à écrire à l'aide de prototype et non en mode Javascript 6.

### A) Définition de la classe et du constructor

Définissez la classe *ObjetQuiTombe* où le constructeur prend un objet comme seul paramètre.

Cet objet doit avoir les propriétés *taille*, *urlImage*, *x*, *debut* et *fin*.

Certaines propriétés ont les valeurs par défaut suivantes :

- *taille* >> {largeur : 30, hauteur : 30}
- *urlImage* >> "missile.jpg"
- *x* >> 200,

Debut et fin sont les coordonnées y de début et de fin de la chute.

Indiquez que cette classe hérite d'*ImageDeplacable* (toujours pas de Javascript 6).

Le constructeur appelle le constructeur de sa classe mère (avec les bons paramètres).

### B) Tombe

Implémenter la méthode *tombe* de la classe *ObjetQuiTombe* qui ne prend aucun paramètre.

La méthode *tombe* va faire tomber l'image (de x,debut à x,fin).

Méthode utile : *setTimeout* (F,T) déclenche la fonction F après avoir attendu T millisecondes. Ici 5ms par pixel est un bon rythme.





## Exercice n°3 : Canon

Cette classe est à écrire à l'aide du mode Javascript 6.

### A) Définition de la classe et du constructor

Définissez la classe *Canon* où le constructeur prend en paramètre *x*, *y* et *fin*.

Le constructeur va créer une *ImageDeplacable* avec les arguments suivants :

- largeur = 22
- hauteur = 47
- urlImage = « canon.jpg »
- x = x
- y = y

Ensuite, le constructeur invoque la méthode *demarre* (voir plus bas)

### B) Demarre

La méthode *demarre* envoie un missile à partir des coordonnées *x,y* jusqu'à *x,fin* toutes les N secondes. N est un nombre au hasard entre 0 et 6 qui change à chaque fois.

Méthode utile : `setTimeout (F,T)`

Nombre au hasard : `Math.floor(Math.random() * Math.floor(6))`





