

Story-board JMD

Download de <https://github.com/flashline/Atelier-JS-2017>

Vérification préalable :

Q: qui a des bases langage de programmation (ce qu'est une fonction et une variable)

Q: Les débutants ont-ils été sur mes liens doc JS, pour (variables, fonctions, bases js) ?

Q: Editeur de texte préféré ou sublime text ou flashdevelop téléchargé ... est-ce ok ?

Aller sur <http://www.pixaline.net/intra/atelier2017/> pour montrer les 2 progs.

<http://www.pixaline.net/intra/atelier2017/easyShoot/>

Revenir sur

Descendre rapidement le html + css dans l'éditeur.

Ce ne sont pas des langages de prog mais des langages de mise en forme du visuel (texte,image,son,vidéo) ; ils n'ont pas de variable, de conditions if else, de boucles.

Théo :

Grand principe en programmation :

Décomposer un projet en modules plus simples et les modules en fonctionnalités encore plus simples.

Q: Comment peut-on décomposer ce prog en sous-programmes plus simple ?

R: cible/**target** , **canon**, **ball**, capture actions utilisateur/**listener**

On y va avec **target**...

Choisir le nom de la var qui représentera la cible dans js.

1/ **Choisir des noms clairs**, pas $y=ax+b$ mais $chargesTotales=CA*coefCharge+chargeFixe$

2/ Comme l'idée de partage est très répandue chez les programmeurs, et que plusieurs programmeurs de langues différentes, peuvent travailler sur même projet.

De préférence en anglais (mais m'interrompre si un mot est inconnu)

L'anglais est de fait la langue de l'informatique.

Q: Choix du nom ?

R: **targetElem**

faire :

`/// Target`

`var targetElem=get("#target");`

Appeler fonction pour affichage aléatoire.

`TargetRefresh()` ;

Q: Comment on programme des **actions à intervalles réguliers**

R : `setInterval()`

expliquer `setInterval()`

`window.setInterval(targetRefresh,5000);`

Pratique : avant de faire `targetRefresh`, créer et expliquer les func util `get()` et `log()`

`function log (o) {get("#info").innerHTML+=o+"
";}`

`function get(s) {return document.querySelector(s);}`

plus simple que `console.log("bonjour")` et `document.getElementById("aDivId")` ;

Créer targetRefresh()

Q: Quelle fonction JS permet de récupérer des nombres aléatoires ?

R: Math.random() ;

Théo : le fonctionnement de Math.random > renvoi 0 à 0,....

```
function targetRefresh () {  
    var x=Math.floor(Math.random() * (600-20) );  
    var y=Math.floor(Math.random() * 200 );  
    targetElem.style.left=x+"px";  
    targetElem.style.top=y+"px";  
};
```

TEST : Fin du module target on teste et débogue

Module **canon**

créer var (canonElem) et initialiser canonX et canonY

/// Canon

```
var canonElem=get("#canon");
```

```
var canonX=285;
```

```
var canonY=315;
```

on verra plus tard comment créer la fonction qui déplace le canon mais on peut déjà la nommer Q : comment ? R : [canonMove\(\)](#)

Note : On verra le module ball après car on a besoin du module listener pour tester canon

Module **listener** (capture des actions utilisateur.)

Q: Quelqu'un connaît-il comment on écoute les événements en JS.

R: addEventListener()

Théo : Explication détail de document.addEventListener("keydown", onKeyDown);

Faire :

/// Listeners adding

```
document.addEventListener("keydown",onKeyDown);
```

Création de la fonction listener

```
function onKeyDown (e) {  
    if (e.keyCode== ??? ) { // left  
    }  
    else if (e.keyCode== ???) { //right  
    }  
}
```

Théo : Explication de l'objet d'événement (e)

objet passé par js

ayant la propriété keyCode.

Q: Cherchez et trouvez le code correspondant aux flèches et espace.

R: <http://keycode.info/>

```
function onKeyDown (e) {
    if (e.keyCode== 37) { // left
    }
    else if (e.keyCode== 39) { //right
    }
    else if (e.keyCode== 32 ) { //shoot
    }
}
```

Q: Comment n'avoir qu'une seule fonction canonMove pour déplacer dans les 2 sens ?

R: envoyer un paramètre positif OU négatif.

Faire

var CANON_STEP=6 ; // en début de prog

et

```
function onKeyDown (e) {
    //left
    if (e.keyCode==37 ) {
        canonMove(-CANON_STEP);
    }
    //right
    else if (e.keyCode==39 ) {
        canonMove(+CANON_STEP);
    }
    else if (e.keyCode==32 ) {
    }
}
```

Q:

Pourquoi c'est bien d'utiliser une constant comme CANON_STEP plutôt qu'écrire la valeur directement dans la fonction ?

R:

c'est plus facile de modifier des lignes en début de prog plutôt que chercher partout dans un programme. Ex passage de la tva de 17,60 à 19,6 puis à 20

Module **canon** > canonMove() et canonRefresh() ;

faire :

```
function canonMove (step) {
    canonX+=step;
    canonRefresh ();
}
function canonRefresh () {
    canonElem.style.left=canonX+"px";
}
```

Théo : canonElem.style.left = du css

TEST :

Q: Qd débogué, voir ce qui cloche en testant sortie d'écran ?

R: ça déborde !

Module **listeners > onKeyDown**

Faire :

```
function onKeyDown (e) {  
    //left  
    if (e.keyCode==37 && canonX>0) {  
        canonMove(-CANON_STEP);  
    }  
    //right  
    else if (e.keyCode==39 && canonX<600-30) {  
        canonMove(+CANON_STEP);  
    }  
    else if (e.keyCode==32 ) {  
    }  
}
```

TEST : Fin du module canon on teste et débogue

Module **ball**

créer var ballElemProto

/// Ball

var ballProtoElem=get("#ballProto");

var nextBallNum=0;

Q: pourquoi proto ? Quelle particularité a ball par rapport à target et canon ?

R: plusieurs ball en rafale.

Théo : on va créer les <div> à partir de la <div> prototype.

on verra plus tard comment créer la fonctions qui lance une nouvelle ball mais on peut déjà les nommer : Q : comment R : [ballShoot\(\)](#)

faire :

```
function onKeyDown (e) {  
    ....  
    else if (e.keyCode==32 ) {  
        ballShoot() ;  
    }  
}
```

Module **ball** > ballShoot()

Q: Intuitivement, quel problème on a si on veut envoyer une rafale de ball ?

R: On va devoir gérer plusieurs visuels (éléments), plusieurs positions.

Q: Comment résoudre le problème ?

R: Avec Array [] mais mieux avec l'objet Object de js

Théo :

un type de var Object c'est pas de la POO mais on met un doigt de pied quand même dans la POO façon JS.

Un object c'est quand plusieurs variables qu'on appelle propriétés (ou élément) sont regroupées dans une seule variable qui est donc du type Object.

On crée l'objet avec var (comme une autre variable)

ex : var ball={} ;

On accède à une propriété en écrivant <nom de l'objet>.<nom de la propriété>

ex : ball.elem

On va donc mettre dans un object ball avec toutes les propriétés d'une balle.

On va créer un objet ball pour chaque balle.

Et passer cet objet ball successivement à toutes les fonctions.

On démarre la fonction...

Au lieu de setInterval on va utiliser une fonction js plus moderne.

```
function ballShoot () {  
    var ball={};  
    ball.elem=ballProtoElem.cloneNode();  
    ball.x=canonX-5; ball.y=canonY;  
    get("#main").appendChild(ball.elem);  
    ballMove(ball);  
};
```

Théo :

Explication de cloneNode() et appendChild()

Q: Pourquoi et comment donner un id unique à chaque ball.elem

R: c'est la philosophie de html même si ça ne provoque pas d'erreur, l'id d'un élément doit être unique.

```
function ballShoot () {  
    nextBallNum++;  
    var ball={};  
    ball.elem=ballProtoElem.cloneNode();  
    ball.elem.id="ball_"+nextBallNum;  
    ball.x=canonX-5; ball.y=canonY;  
    get("#main").appendChild(ball.elem);  
    ballMove(ball);  
}
```

créer : `var CANON_BALL_SPEED=2;` // en début de prog

Et faire :

```
function ballMove(ball) {  
    ball.y-=CANON_BALL_SPEED ;  
    ballRefresh(ball);  
    var loop=function () { ballMove(ball); }  
    window.requestAnimationFrame(loop);  
}
```

faire :

```
function ballRefresh (ball) {  
    ball.elem.style.left=ball.x+"px";  
    ball.elem.style.top=ball.y+"px";  
};
```

Q: Quel problème on va avoir si on laisse ballMove comme ça ?

Show : Montrer avec l'inspecteur Chrome la création dynamique dans le html des div ball et ainsi le lien étroit entre html et js

R: les balles ne sont plus visibles mais les fonctions tournent tjrs et les objets s'accumulent en mémoire ; le tout indéfiniment !

Il faut donc rajouter :

```
function ballMove(ball) {  
    ball.y-=CANON_BALL_SPEED ;  
    ballRefresh(ball);  
    if (ball.y>-30) {  
        var loop=function (timestamp) { ballMove(ball); }  
        window.requestAnimationFrame(loop);  
    }  
    else get("#main").removeChild(ball.elem);  
}
```

TEST : Fin du module ball on teste et débogue

Théo : Les commentaires ; ni trop peu ; ni trop

Ajouts des commentaires :

- début de modules

- début de fonctions avec description des params en entrée et en sortie

- ça suffit si les var sont bien nommées.

Y a-t-il des questions ?

FIN