

Introduction à la Simulation

ou comment créer votre premier jeu vidéo

Jacques Duma



HTML, CSS, JavaScript : Présentation des principales technologies du Web, des outils informatiques et des langages de programmation disponibles sur les Navigateurs Internet.

Réalisation d'un petit jeu interactif très simple pour Navigateurs Internet :

La Balle Rebondissante

Cet exemple sera développé intégralement et les codes sources fournis.

A la fin de cette conférence, vous saurez dans les grandes lignes ce qu'il faut faire pour créer une page Web animée et interactive. Vous aurez un aperçu des langages informatiques de base pour cela : HTML, CSS, JavaScript. Vous aurez vu un exemple complet que vous pourrez refaire chez vous. Vous aurez des références pour continuer à approfondir tout cela si vous le souhaitez.

Table des matières

1	Page d'un Navigateur Internet	2
1.1	Aspect visuel souhaité	2
1.2	Code source de la page HTML	2
1.3	Feuille de Styles en CSS	3
1.4	Résumé de ce qui a été vu	4
2	Simulation : Points, Vecteurs, Symétrie, Temps, Vitesse	5
2.1	Animation	5
2.2	Rebonds de la balle sur les murs horizontaux	6
2.3	Rebonds de la balle sur les murs verticaux	6
2.4	Résumé de ce qui a été vu	6
3	Programmation de l'animation : Langage JavaScript	7
3.1	Animation : Balle Rebondissante	7
3.2	Modification du source HTML	9
3.3	Interaction de l'utilisateur : Jouer à attraper la Balle	9
3.4	Résumé de ce qui a été vu	11
4	C'est simple : Vous pouvez le faire !	11
4.1	Édition des codes sources	11
4.2	Exécution du programme	12
4.3	Informations complémentaires	12

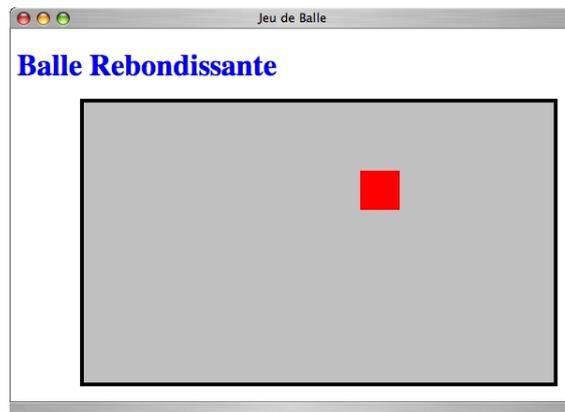
1 Page d'un Navigateur Internet

1.1 Aspect visuel souhaité

Quand on utilise un Navigateur Internet, on émet des *requêtes* pour *charger* les pages de texte que l'on désire consulter.

Ces pages de texte son *codées* en HTML (HyperText Markup Language)

Aspect visuel souhaité



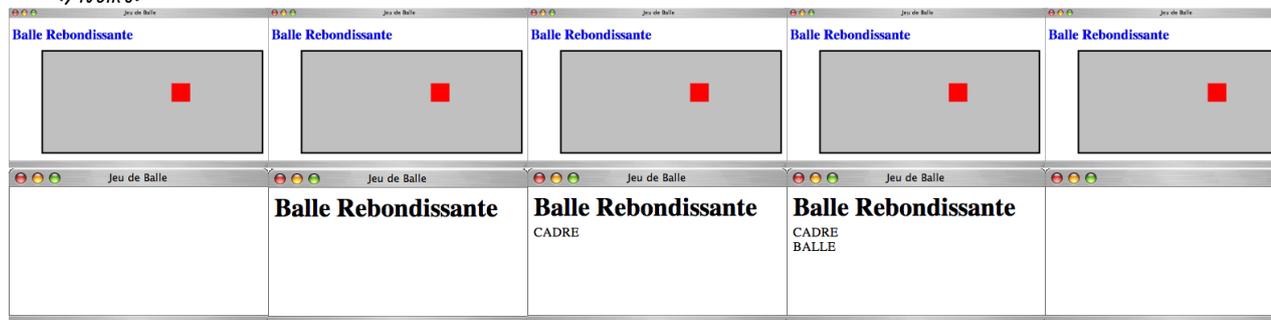
1.2 Code source de la page HTML

Le contenu du document est structuré à l'aide de *balises*.

Page HTML :

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
  </head>
  <body>
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      CADRE
    <div>BALLE</div>
  </div>
</body>
</html>
```



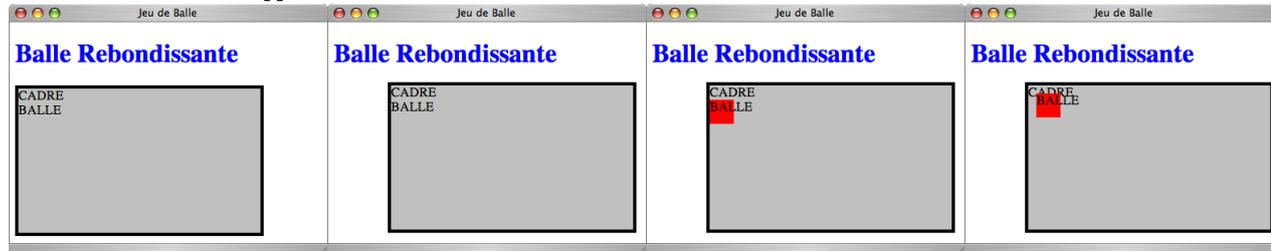
1.3 Feuille de Styles en CSS

Feuille de styles en CSS :

«style.css»

```
h1 { color : blue; }
#cadre {
    width : 500px; height : 300px;
    border : 4px black solid;
    background-color : silver;
    position : fixed;
    top : 75px; left : 75px;
}
#balle {
    width : 100px; height : 100px;
    background-color : red;
    position : absolute;
    top : 10px; left : 10px;
}
```

CADRE et BALLE supprimés du texte HTML



Dans les définitions ci-dessus, l'attribut `position` permet de fixer l'origine du système de coordonnées :

- si `position : fixed`; le point gauche-haut de la fenêtre
- si `position : absolute`; le point gauche-haut de l'élément père
- sinon, par défaut à la suite de ce qui précède dans le flot du texte

Il est ainsi possible de modifier l'aspect graphique de l'affichage en définissant des styles pour chaque élément du document à l'aide du langage CSS.

Lien entre HTML et CSS :

```
<html>
<head>
  <title>Jeu de Balle</title>
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
  <h1>Balle Rebondissante</h1>
  <div id="cadre">
    CADRE
    <div id="balle"> BALLE </div>
  </div>
</body>
</html>
```

Cascading Style Sheets : Feuille de style en cascade.

Des documentations sur [HTML](#) et [CSS](#) sont disponibles en ligne sur Internet.

1.4 Résumé de ce qui a été vu

Le HTML ou *Langage à Balises pour l'Hyper-Texte*.

HTML : HyperText Markup Language

Balises pré-définies (tag)

html, head, link, body, h1, div, ... etc.

Structure : Texte écrit entre deux balises de même nom

```
<html> balise ouvrante
... ici du Texte éventuellement encadré par d'autres balises ...
</html> balise fermante
```

Liste Attribut-Valeur dans les balises ouvrantes

```
<div id="balle">
```

En HTML, il existe un certain nombre de *balises* pré-définies comme la balise `<html>` qui délimite le document en entier, la balise `<div>` pour les divisions ou la balise `<p>` pour les paragraphes.

Les balises `<h1>` ... `<h6>` permettent de structurer sur 6 niveaux, les textes du document en sections, sous-sections, etc.

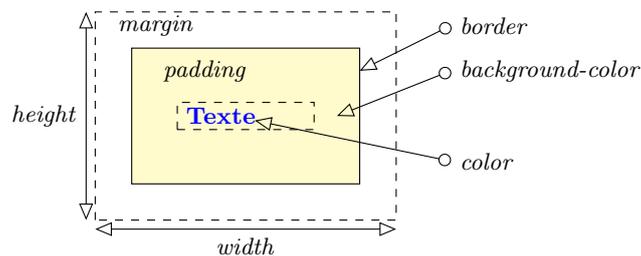
La page que l'on veut créer contient :

- un Titre entre les balises `<h1>`
- le Cadre du jeu dans une balise `<div>`
- la Balle dans une balise `<div>` placée dans la division du Cadre

Pour être plus tard accessibles par programme, ces deux `<div>` sont identifiées respectivement par les *attributs* `id="cadre"` et `id="balle"`.

CSS : Cascading Style Sheets

Géométrie des cases de texte : Le texte de chaque élément de la structure est affiché dans un rectangle paramétrable :



```
#balle { width: 100px; height: 100px; background-color: red;
position: absolute; top: 10px; left: 10px; }
```

Par défaut, `margin` et `padding` sont nuls, `color` est noir et `background-color` est blanc. Les valeurs de `width` et `height` sont calculées automatiquement pour enserrer le contenu de la balise.

Dans la feuille de style CSS les éléments du document sont repérés par :

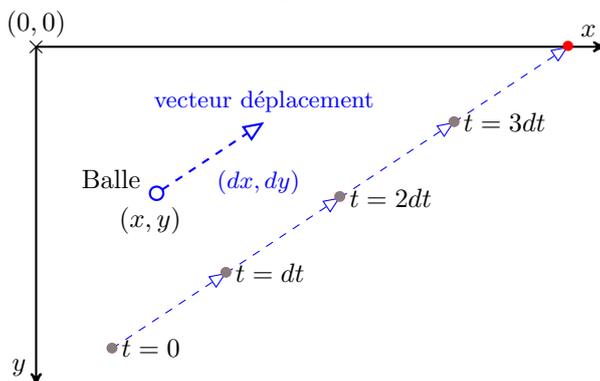
- un *nom simple* qui fait référence à un nom de *balise* ou **tag**
- un *nom précédé de #* qui fait référence à un *identificateur* : attribut **id**
- un *nom précédé de .* qui fait référence à une *classe* : attribut **class**

2 Simulation : Points, Vecteurs, Symétrie, Temps, Vitesse

2.1 Animation

Animation : Affichage image par image

Une Animation consiste à changer périodiquement l'image affichée



Attention : L'axes des abscisses est normalement orienté de gauche à droite, mais l'axes des ordonnées est orienté vers le bas contrairement à l'usage mathématique habituel.

Ainsi, la position du point est repérée par ses coordonnées (x, y) et son déplacement est modélisé à l'aide du vecteur de coordonnées (dx, dy)

Basée sur la persistance rétinienne, l'animation de type cinématographique, et obtenue en affichant des images modifiées plusieurs fois par seconde, ce qui donne l'illusion du mouvement.

L'illusion se produit à partir d'une dizaine d'images par seconde. Le cinéma utilise de façon usuelle 24 images par seconde.

Dans le cinéma expérimental de très haute définition, on améliore le réalisme par des animations beaucoup plus fluides avec 50 à 100 images par seconde.

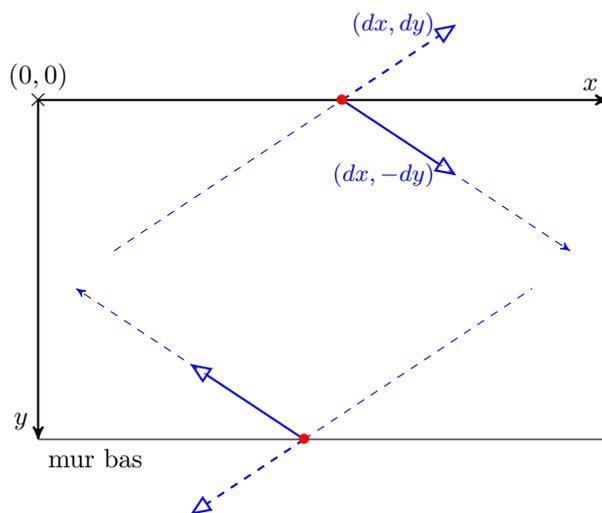
Sur ordinateur la vitesse de calcul permet de réactualiser l'image à haute fréquence, mais il faut limiter la vitesse de rafraîchissement de l'image à ce que l'œil humain est capable de percevoir.

Le choix d'un $dt = 20$ milli secondes (50 images par secondes) donne une animation fluide et laisse le temps à l'utilisateur d'attraper la balle.

2.2 Rebonds de la balle sur les murs horizontaux

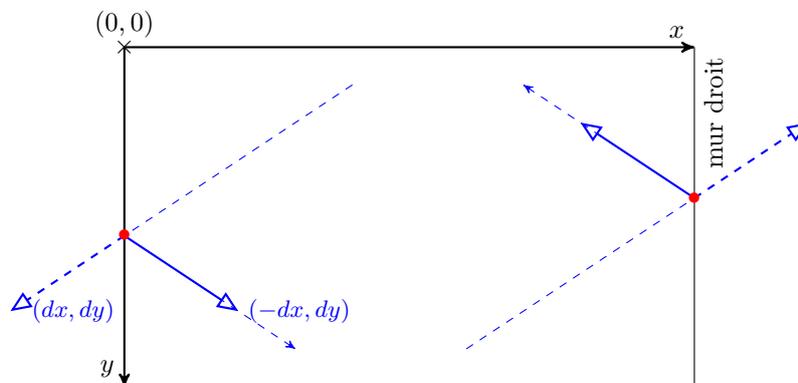
On constate alors que les rebonds peuvent être interprétés comme des symétries axiales appliquées au vecteur de déplacement.

Symétrie par rapport à un axe horizontal



2.3 Rebonds de la balle sur les murs verticaux

Symétrie par rapport à un axe vertical



On constate que le changement de direction de la balle lorsqu'elle touche un mur, correspond à une symétrie de son vecteur de déplacement par rapport à l'axe de ce mur.

2.4 Résumé de ce qui a été vu

Les mathématiques sont nécessaires pour modéliser la situation, mais un peu de physique est maintenant utile pour modéliser la simulation du mouvement.

Mouvement

Le mouvement de la balle sur la page sera simulé par un affichage image par image

Temps

Le temps écoulé entre deux images sera le nombre dt

Vitesse

La vitesse est une variation de *position* par rapport au *temps* le *déplacement* sera représentée par le vecteur (dx, dy)

Position, Vitesse, Temps, Mouvement

Position

La *position* est représentée par les coordonnées (x, y)

3 Programmation de l'animation : Langage JavaScript

L'ordinateur permet d'introduire le temps pour effectuer l'animation.

3.1 Animation : Balle Rebondissante

JavaScript : Initialisation

«code.js»

```
var div_balle; // <div> de la balle
var timer;    // pour l'animation avec setInterval

function avancerBalleDunPas() {
    . . .
}

// initialisation lors du chargement de la page HTML
function init() {
    div_balle = document.getElementById("balle");
    timer = setInterval(avancerBalleDunPas, dt);
}
```

Les coordonnées (x, y) repèrent la position (point gauche-haut) de la balle dans le cadre de jeu, exprimée en pixels.

Le vecteur (dx, dy) exprime en pixels, un pas de déplacement de la balle.

Le mouvement de la balle est limité par le cadre :

- horizontalement entre 0 et murDroit
- verticalement entre 0 et murBas

Ces deux constantes tiennent compte de la largeur de la balle, pour que celle-ci ne franchisse pas la limite des murs, à droite et en bas.

L'animation est contrôlée à l'aide de dt le nombre de milli secondes qui s'écoule entre chaque rafraîchissement de l'image dans la page.

la *fonction JavaScript* `document.getElementById()` récupère l'unique élément du document repéré par l'attribut `id="balle"` et on mémorise cet élément dans la *variable* `div_balle`.

la *fonction JavaScript* `setInterval` programme l'exécution périodique d'une fonction JavaScript (ici `avancerBalleDunPas`) à intervalle régulier (ici chaque *dt* milli secondes). On mémorise l'objet JavaScript responsable de cette tâche dans la *variable* `timer`, pour un usage qui sera vu ultérieurement.

JavaScript : Animation de la Balle

«code.js»

```
// appel tous les dt milli secondes
function avancerBalleDunPas() {
    x = x + dx;
    y = y + dy;
    if (x < 0 || x > murDroit) {
        dx = -dx; // rebond horizontal
    } else if (y < 0 || y > murBas) {
        dy = -dy; // rebond vertical
    } else { // on change la position de la balle
        div_balle.style.left = x + "px"; // left : 5px
        div_balle.style.top = y + "px"; // top : 9px
    }
}
```

La fonction JavaScript `avancerBalleDunPas` est exécutée chaque *dt* milli secondes. Elle calcule les nouvelles coordonnées de la Balle en tenant compte des rebonds éventuels, avant de modifier la position de celle-ci.

Le changement de position est possible à l'aide de la propriété `style` de tout élément du document. Il suffit de modifier les valeurs de `style.left` et `style.top`

Ainsi par exemple si $(x, y) = (25, 55)$ les commandes JavaScript `div_balle.style.left = x + "px";` `div_balle.style.top = y + "px";` produisent le même effet que la définition CSS suivante :

```
#balle { left : 25px; top : 55px }
```

On rappelle que dans la feuille de style CSS il a déjà été précisé, pour l'élément qui représente la Balle : `position : absolute;` ce qui signifie que les coordonnées de la balle sont exprimées dans le repère du Cadre.

JavaScript : Déclaration des variables

«code.js»

```
var div_balle; // <div> de la balle
var timer; // pour l'animation avec setInterval

var x = 250; // balle au centre du cadre
var y = 150; // 1/2*(largeur_cadre, hauteur_cadre)

var dx = 5; // vecteur du mouvement de la balle
var dy = 2; // par exemple (5, 2)

var murDroit = 500 - 100; // largeur_cadre - largeur_balle
var murBas = 300 - 100; // hauteur_cadre - hauteur_balle

var dt = 20; // temps en milli secondes entre chaque pas
```

3.2 Modification du source HTML

HTML : Appel JavaScript

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle"></div>
    </div>
  </body>
</html>
```

La valeur de l'attribut `onload` est une commande JavaScript qui sera exécutée dès que la page HTML est chargée dans le navigateur Internet.

3.3 Interaction de l'utilisateur : Jouer à attraper la Balle

Pour l'instant la Balle se contente simplement de rebondir indéfiniment entre les quatre murs du Cadre.

On veut que l'utilisateur puisse immobiliser la Balle en faisant un Clic avec la souris. Le problème est donc de prendre en compte les événements extérieurs au programme déclenchés par l'utilisateur sur l'interface graphique.

Pour cela, il va encore falloir modifier le code HTML de la page. On va ajouter deux attributs supplémentaires à la liste de la balise `<div id="balle">`

HTML : Clic sur la Balle

«index.html»

```
<html>
  <head>
    <title>Jeu de Balle</title>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="code.js" type="text/javascript"></script>
  </head>
  <body onload="init()">
    <h1>Balle Rebondissante</h1>
    <div id="cadre">
      <div id="balle" class="mobile" onclick="clicBalle()"></div>
    </div>
  </body>
</html>
```

La valeur de l'attribut `onclick` de la balise `<div id="balle">` définit la commande JavaScript à exécuter si l'utilisateur effectue un Clic sur la représentation graphique de l'élément du document correspondant. Il ne reste plus qu'à définir la fonction `clicBalle()` dans le programme JavaScript.

La valeur de l'attribut `class` de la balise `<div id="balle">` associe un nom de classe à l'élément du document correspondant. Le langage CSS permet de définir le style d'un élément selon sa classe, il sera donc ainsi possible de modifier l'aspect visuel d'un élément en modifiant la feuille de style.

CSS : Modification de la Balle

«style.css»

```
#balle {  
    position : absolute;  
    width : 200px; height : 200px;  
}  
  
.mobile { background-color : red; }  
  
.immobile {  
    background-color : yellow;  
    border : 3px black dashed;  
}
```



Remarquons d'abord que le style pour l'élément `<div id="balle">` a été simplifié. On précise simplement que les coordonnées de la Balle sont exprimées dans le repère du Cadre.

Par contre nous définissons des styles différents pour les deux classes `mobile` et `immobile` qui seront attribuées à la Balle.

Un élément du document ne peut avoir qu'un identificateur (attribut `id`), non modifiable, mais sa classe (attribut `class`) peut être changée par programme.

Dans le document HTML, un identificateur ne peut être donné qu'à un élément unique, mais plusieurs éléments différents peuvent avoir la même classe et donc partager un même style.

JavaScript : Réaction de la Balle

«code.js»

```
// appel lors d'un clic sur la balle  
function clicBalle() {  
    if (div_balle.className === "mobile") {  
        clearInterval(timer); // Stop  
        div_balle.className = "immobile";  
    } else {  
        timer = setInterval(avancerBalleDunPas, dt); // Go  
        div_balle.className = "mobile";  
    }  
}
```

La valeur de l'attribut `class` est accessible en lecture et en écriture à l'aide de la propriété `className` de tout élément du document.

La commande `clearInterval(timer)` ordonne au `timer` d'interrompre les appels à la fonction `avancerBalleDunPas` qu'il effectue périodiquement, ce qui stoppe l'animation et fige la Balle.

Le changement de classe de l'élément permet de modifier l'aspect visuel de la Balle selon qu'elle bouge ou non.

Attention : Un navigateur Internet n'est pas une console de jeu de dernière génération. Les interactions de l'utilisateur avec l'interface graphique sont assez lentes. On pourra modifier la valeur de la variable `dt` pour modifier la vitesse de la Balle et rendre le jeu confortable.

De nombreuses documentations sur [HTML](#) sur [CSS](#) et sur [JavaScript](#) sont disponibles en ligne sur Internet.

Pensez à utiliser Google pour obtenir des informations.

Clic sur les liens : [HTML](#) , [CSS](#) ou [JavaScript](#) pour des sites simples, intéressants pour débiter.

3.4 Résumé de ce qui a été vu

HTML CSS JavaScript

Tout Navigateur Internet interprète :

HTML

Langage de description du contenu textuel des pages

CSS

Langage de description de l'aspect visuel de l'affichage

JavaScript

Langage de programmation des interactions du client sur l'inteface

4 C'est simple : Vous pouvez le faire !

4.1 Édition des codes sources

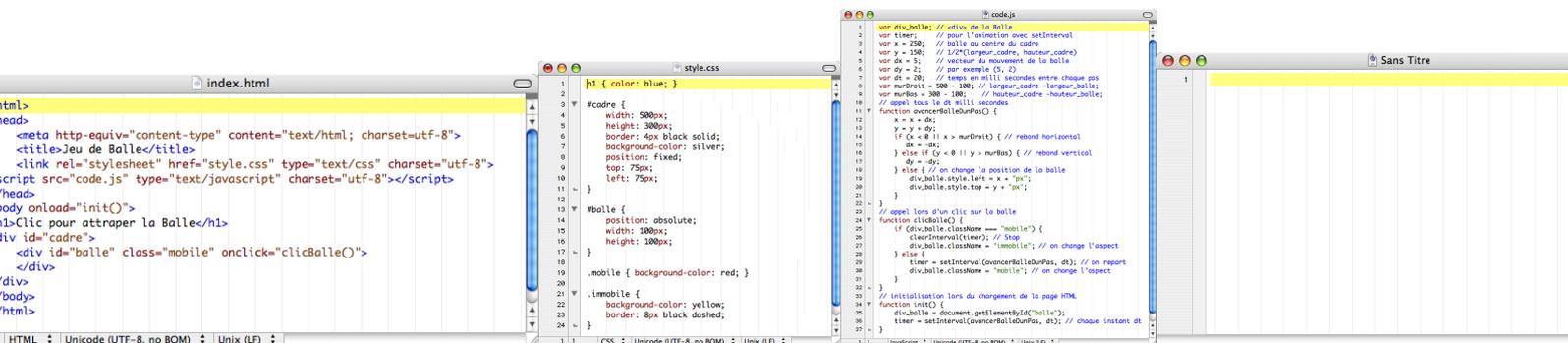
Sur un ordinateur, tout le monde dispose d'un éditeur de texte et d'un Navigateur Internet.

Donc : tout le monde peut réaliser le petit exemple proposé ci-dessus.

Cet exemple a volontairement été conçu pour être le plus simple possible. Ceci dans le but d'être effectivement réalisable, même par un débutant.

Vous pouvez le faire : Édition des codes sources

[Il vous suffit d'un éditeur de texte pour créer trois fichiers](#)



- «index.html» la page HTML
- «style.css» la feuille de style CSS
- «code.js» le code JavaScript

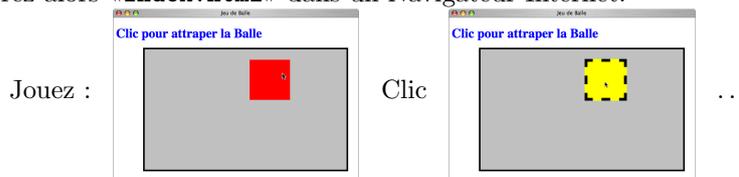
4.2 Exécution du programme

Vous pouvez le tester : Exécution du programme

- Placez ces 3 fichiers dans un même dossier : «**Balle**»



- Ouvrez alors «**index.html**» dans un Navigateur Internet.



Ensuite... donnez libre cours à votre imagination ...

Dès qu'on a réussi à faire fonctionner ce petit exemple, on peut chercher sur Internet des informations pour continuer. De très nombreuses documentations sur HTML, CSS et JavaScript sont disponibles en ligne.

4.3 Informations complémentaires

Pour plus d'informations :

Vous trouverez les exemples présentés dans cet exposé sur le site :

Mathématiques et Informatique

<http://math.et.info.free.fr/>

... ainsi que d'autres exemples et des informations complémentaires

Et aussi de nombreux problèmes et exemples sur le site :

Atelier de Mathématiques «Laurent Schwartz»

<http://ateliermathematique.free.fr/>



On trouvera de nombreuses informations sur les sites indiqués ci-dessus.

Il est aussi possible de me contacter, pour demander éventuellement des informations complémentaires, en remplissant les formulaires disponibles sur chacun de ces sites.