

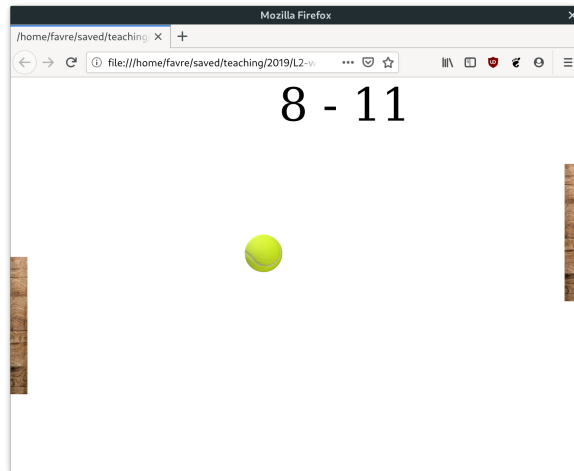
# TP Web : Un pong en HTML/javascript

Benoit Favre, modifié par Thomas Schatz

généré le 24 janvier 2022

## 1 Introduction

L'objectif de cette séance est de créer un jeu simple (pong) à l'aide de HTML, CSS et javascript.



## 2 Page

Créer la page HTML « pong.html » du jeu qui contiendra les éléments suivants :

- Une image de la balle d'identifiant 'ball' (par exemple celle-ci)
- Une image pour la raquette de chaque joueur d'identifiant 'paddle1' et 'paddle2' (par exemple celle-là)
- Une div pour les scores des joueurs, d'identifiant 'scores'

La page fera aussi référence à un fichier `pong.css` et un fichier `pong.js` contenant respectivement la feuille de style CSS et le javascript du jeu.

## 3 CSS

Écrire la feuille de style correspondant aux éléments suivants :

Le CSS devra tout d'abord définir la taille des différents éléments : 64×64 pixels pour la balle et 24×192 pour les raquettes.

Ensuite, l'idée est de placer les éléments du jeu à des positions fixes (`position: fixed`) et de manipuler leur position grâce aux propriétés CSS `left` (abscisse) et `top` (ordonnées) exprimées en pixels (px). Pour pouvoir le faire, nous allons définir le corps de la page (body) comme aire de jeu.

Une fonction javascript permettra de récupérer sa taille et donc de jouer quelle que soit la taille de la fenêtre. Le CSS correspondant permet à la balise body de remplir la fenêtre.

```
body {  
  position: fixed;  
  top: 0px;  
  left: 0px;  
  right: 0px;  
  bottom: 0px;  
  margin: 0px;  
}
```

Les scores seront en haut au centre de la page avec une taille suffisante pour que l'on puisse les voir.

## 4 Échauffement : faire bouger la balle

Voici un exemple simple qui permet de faire bouger la balle à l'écran. On a tout d'abord un constructeur `Ball` pour l'objet balle. Ce dernier renvoie une balle avec des coordonnées `x` et `y`, et l'identifiant de l'élément HTML la représentant. La fonction `place_objects` prend en paramètre une liste d'objets et met à jour la position des éléments HTML qui les représentent. La fonction `update` est celle dans laquelle on implémentera le cœur du jeu. Elle est appelée toutes les 100ms et met à jour la position des objets, puis place les éléments HTML correspondant. Cette architecture permet de séparer la représentation des objets et leur affichage. La fonction `init` initialise le jeu en donnant une valeur à la variable globale `ball` et met en place l'appel en boucle de la fonction `update` grâce à `setInterval`.

```
function Ball() {  
  this.id = "ball";  
  this.x = 0;  
  this.y = 0;  
}  
  
function place_objects(objects) {  
  for(let object of objects) {  
    let element = document.getElementById(object.id);  
    element.style.left = object.x + "px";  
    element.style.top = object.y + "px";  
  }  
}  
  
function update() {  
  ball.x += 5;  
  ball.y += 5;  
  
  place_objects([ball]);  
}  
  
let ball;  
  
function init() {  
  ball = new Ball();
```

```
setInterval(update, 100);
}
```

## 5 Faire rebondir la balle

Pour faire rebondir la balle, on a besoin de deux choses : la direction de déplacement de la balle (c'est à dire le nombre de pixel qu'il faut ajouter à ses coordonnées à chaque appel de `update`), et la taille de l'espace de jeu, donc de la balise `body`.

Pour le premier, ajoutez deux champs au constructeur `Ball` : `vx` et `vy` qui représentent la vitesse (vitesse de déplacement) dans les deux directions. Initialisez-les à des valeurs non-nulles et utilisez-les à la place de `(5, 5)` dans `update`.

Pour le second, utilisez `document.body.getBoundingClientRect()` qui renvoie un objet javascript avec les champs : `x`, `y`, `width`, `height`.

Implémentez les rebonds dans `update` de la façon suivante :

- Si l'abscisse de la balle devient négatif, ou devient plus grand que la largeur du `body` plus la largeur de la balle, alors inversez `vx`.
- Si l'ordonnée de la balle devient négative, ou devient plus grande que la hauteur du `body` plus la hauteur de la balle, alors inversez `vy`.

Une fois ces règles mises en place, la balle devrait rebondir sur les côtés de la fenêtre.

## 6 Placement et contrôle des raquettes

Créez un constructeur pour les raquettes. Ces dernières devront être placées à gauche et à droite de l'espace de jeu (donc en fonction de la taille de celui-ci). On veut que les deux joueurs puissent les contrôler en appuyant sur une touche pour aller vers le haut ou vers le bas. Les raquettes devront s'arrêter lorsqu'elles atteignent le bord de l'espace de jeu.

Pour récupérer les appuis sur les touches, on définira un objet global `buttons` contenant quatre booléens permettant de savoir si les touches correspondant aux actions des joueurs sont appuyées : `p1_up`, `p1_down`, `p2_up`, et `p2_down`. La mise en place de deux traqueurs d'évènements (`keydown` et `keyup`) avec `document.addEventListener` et comme fonction de rappel (callback) la fonction `track_player_input`, comme indiqué dans le code ci-dessous, permet de mettre à jour la valeur de ces booléens lorsque les joueur appuient sur les touches<sup>1</sup> 'a', 'q', 'p' et 'm'.

```
function track_player_input(event) {
  if(event.type == "keydown") {
    switch(event.key) {
      case "a": buttons.p1_up = true; break;
      case "q": buttons.p1_down = true; break;
      case "p": buttons.p2_up = true; break;
      case "m": buttons.p2_down = true; break;
    }
  } else if(event.type == "keyup") {
    switch(event.key) {
      case "a": buttons.p1_up = false; break;
      case "q": buttons.p1_down = false; break;
      case "p": buttons.p2_up = false; break;
      case "m": buttons.p2_down = false; break;
    }
  }
}
```

---

1. On peut aussi utiliser les flèches mais il faut pour cela utiliser `event.keyCode` avec le numéro de la touche indiqué dans la documentation HTML.

```
}  
document.addEventListener("keydown", track_player_input);  
document.addEventListener("keyup", track_player_input);
```

Mettez la position des raquettes à jour en fonction de l'état des boutons dans la fonction `update`.

## 7 Rebondir sur les raquettes

Modifiez la fonction `update` pour que les rebonds à droite et à gauche ne soient possibles que si la balle touche la raquette, c'est à dire que leurs rectangles sont en intersection. Pour simplifier, pour la raquette de gauche, il suffit de tester si la balle a une abscisse inférieure à la largeur de la raquette et une ordonnée entre le haut et le bas de la raquette, après avoir ajouté la moitié de la hauteur de la balle. Notez que cette simplification ne marche que si la vitesse de la balle est moins grande que la largeur de la raquette.

## 8 Mettre à jour le score et commencer une autre manche

Ajouter une variable globale contenant le score de chaque joueur. Mettre à jour le contenu de la div `#scores` à l'aide de ces scores dans `update`. Si la balle sort complètement de l'écran à droite ou à gauche, ajouter 1 au score du joueur concerné, remettre la balle au milieu et donnez lui une direction aléatoire<sup>2</sup>.

---

2. Vous pouvez tirer aléatoirement un angle entre  $[-\frac{\pi}{4}, \frac{\pi}{4}]$  ou  $[\frac{3\pi}{4}, \frac{5\pi}{4}[$ , et calculer  $v_x$  et  $v_y$  en fonction de cet angle.