

EXPLICATIONS DE L'ANIMATION D'UN JEU DE TENNIS AVEC CANVAS / JAVASCRIPT



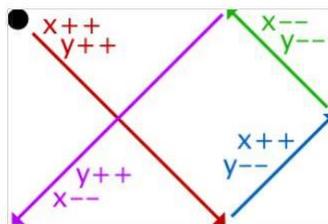
On souhaite élaborer une animation graphique évoluée dans laquelle :

- une balle rebondit sur les bords du cadre rectangulaire ;
- une raquette, contrôlée par les flèches « ← » et « → » du clavier permet de renvoyer la balle vers le haut ;
- une fenêtre pop-up apparaît lorsque le jeu est stoppé, c'est-à-dire lorsque la balle passe en dessous de la raquette et vient toucher le bas du terrain.

1. Animation d'une balle dans un rectangle

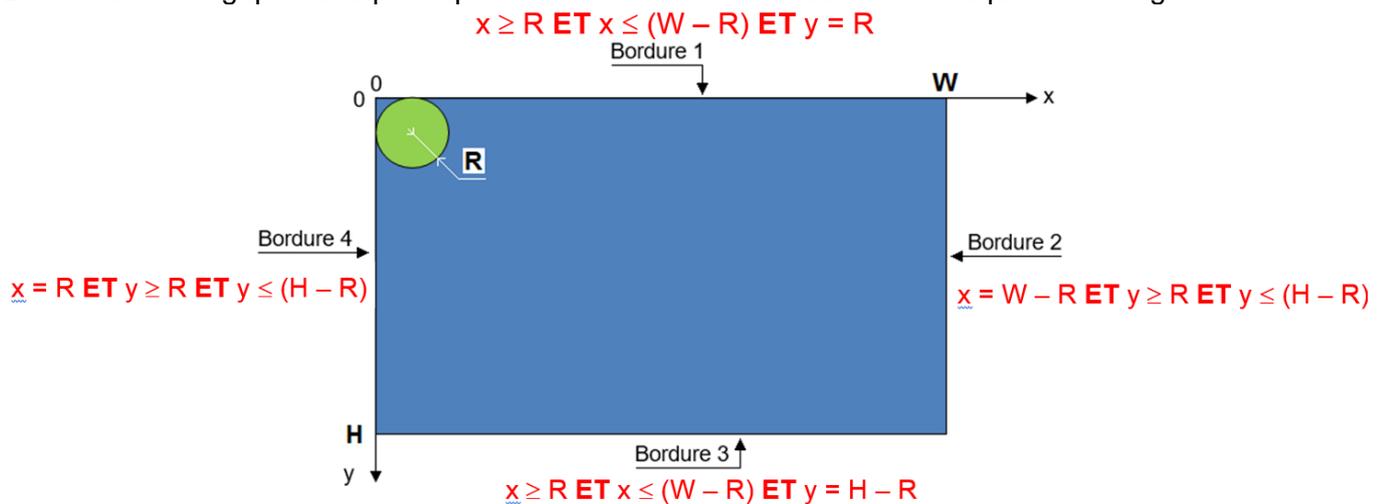
Objectif : réaliser une animation dans laquelle la balle rebondit sur les bords du cadre.

Le sens des incréments (positifs, négatifs) en x et en y est rappelé ci-dessous.



On donne : R : Rayon de la balle, W : Largeur du terrain, H : Hauteur du terrain

Les conditions logiques indiquant que la balle touche une bordure sont indiquées en rouge :



On élabore maintenant un algorithme à partir des conditions initiales ci-dessous.

Variable globale	Signification	Unité	Condition initiale
X	Position en x du centre de la balle	pixel	6
Y	Position en y du centre de la balle		6
W	Largeur max		170
H	Hauteur max		150
R	Rayon de la balle		6
a	Incrément du déplacement en x (entier)		1
b	Incrément du déplacement en y (entier)		1

Le déplacement de la balle et des raquettes se fera dans une fonction « animation » qui sera effectuée 30 fois par seconde par le programme. Son algorithme est le suivant :

fonction animation

```

DEB
  x = x + a // déplacement horizontal
  y = y + b // déplacement vertical
  SI x ≥ R et x ≤ W - R // si la balle ne tape pas les bordures gauche et droite
  DEB
    SI y ≤ R // Rebond bordure 1 (haute)
    | alors b = 1 // déplacement vers le bas
    FINSI
    SI y ≥ (H - R) // Rebond bordure 3 (basse)
    | alors b = -1 // déplacement vers le haut
    FINSI
  FIN
  SI y ≥ R et y ≤ (H - R) // si la balle ne tape pas les bordures haute et basse
  DEB
    SI x ≤ R // Rebond bordure 4 (gauche)
    | alors a = 1 // déplacement vers la droite
    FINSI
    SI x ≥ (W - R) // Rebond bordure 2 (droite)
    | alors a = -1 // déplacement vers la gauche
    FINSI
  FIN
FIN

```

Pour rendre les rebonds plus crédibles, les incréments peuvent être générés de manière aléatoire à chaque rebond.

La fonction **getRandom(min, max)** est donnée ci-dessous.

La fonction **Math.random()** renvoie un flottant (chiffre à virgule) compris dans l'intervalle [0 , 1[.

```

function getRandom(min, max) { // On renvoie un ENTIER aléatoire
  min=Math.ceil(min); // compris entre min et max
  max=Math.floor(max);
  return Math.floor(Math.random() * (max-min) ) +min;
}

```

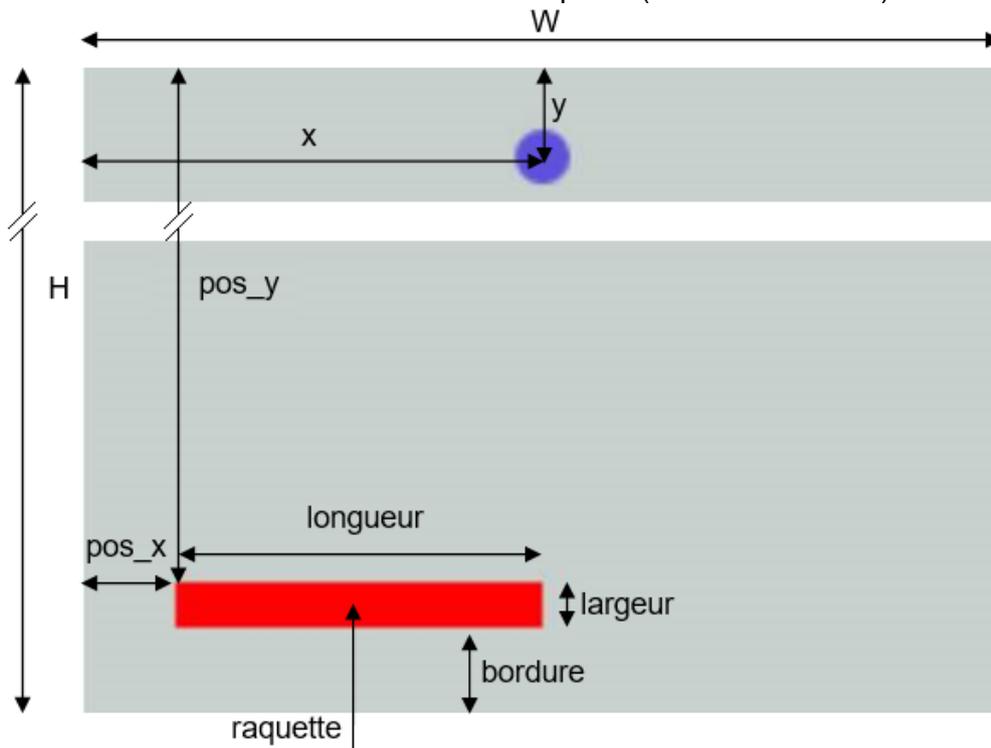
2. Mise en place d'une raquette - Gestion du déplacement par le clavier

Les nouvelles variables concernant la raquette sont :

Variable globale	Signification	Unité	Condition initiale
pos_x	Position en x de la raquette	pixel	0
pos_y	Position en y de la raquette		H-largeur-bordure
longueur	Longueur de la raquette		80
largeur	Largeur de la raquette		10
bordure	Distance entre la bas de la raquette et le bas du terrain		20

Remarque : Pour la raquette, seule la position en x « **pos_x** » va varier dans l'exécution du programme

Illustrations des variables concernant la raquette (et la balle en bleu) :



Remarque : Les dimensions du terrain sont W et H

EXEMPLE DE CODE :

```

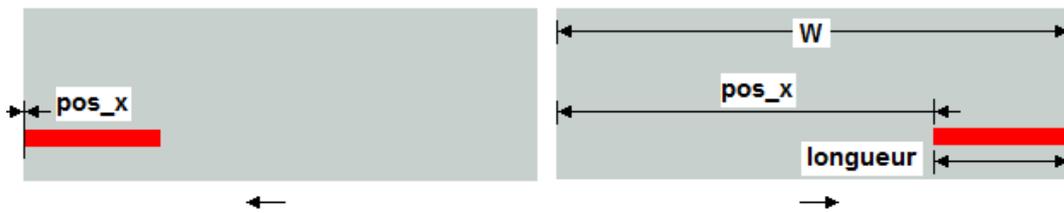
// Variables liées à la raquette
var largeur=10;
var longueur=80;
var pos_x=0; // position en x de la raquette
var bordure=20; // bordure de bas de page
var pos_y=H-largeur-bordure; // position en y de la raquette

function animation(){

// Raquette (rectangle)
context.fillStyle="rgb(255,255,0)"; // couleur de la raquette
context.fillRect(pos_x,pos_y,longueur,largeur); // position et taille

```

CONDITIONS sur la variable « pos_x » pour que la raquette ne déborde ni à gauche ni à droite du terrain (rectangle gris).



Déplacement vers la gauche (←)

$$\text{pos_x} \geq 0$$

Déplacement vers la droite (→)

$$\text{pos_x} \leq W - \text{longueur}$$

Exemple de code pour la fonction de lecture du clavier permettant d'identifier l'appui sur les touches « flèche gauche » : n° 37 et « flèche droite » : n° 38

```
// Définitions de la lecture des touches du clavier
function lect_clavier(evt) {
  switch (evt.keyCode) {

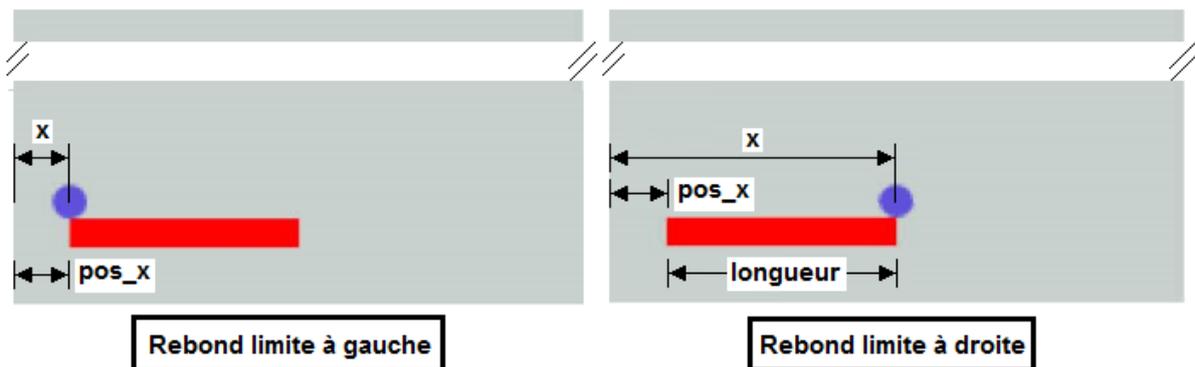
    case 37: // fleche gauche
      pos_x=pos_x-10; // déplacement à gauche de 10 pixel
      if(pos_x<=0)pos_x=0; // arret du déplacement si position négative
      break;

    case 39: // flèche droite
      pos_x=pos_x+10; // déplacement à droite de 10 pixel
      if(pos_x>=(W-longueur))pos_x=W-longueur; // arret du déplacement si position hors terrain
      break;

  }
}
```

3. Rebond de la balle sur la raquette

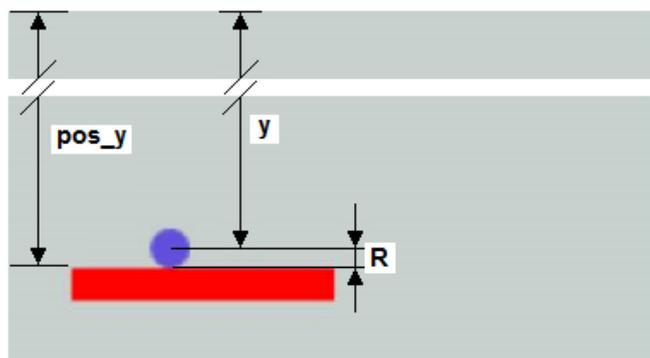
On donne des conditions limites pour que la balle puisse rebondir sur la raquette :



Un rebond limite à **gauche** se fera lorsqu'on aura la condition suivante : $x \geq \text{pos_x}$

Un rebond limite à **droite** se fera lorsqu'on aura la condition suivante : $x \leq \text{pos_x} + \text{longueur}$

Un rebond sur le rebord du **haut de la raquette** se fera lorsqu'on aura la condition suivante : $y = \text{pos_y} - R$



La balle va faire un rebond sur la raquette, vers le haut ($B = -1$) si la balle touche le haut de la raquette ($y = \text{pos_y} - R$) ET ne déborde ni à gauche de la raquette ($x \geq \text{pos_x}$) ni à droite de la raquette ($x \leq \text{pos_x} + \text{longueur}$).

Cela se traduit par la ligne de code suivante :

```
If (y==pos_y-R && x>=pos_x && x<=pos_x + longueur) B=-1 ;
```

Remarque : dans un test (If) l'égalité s'écrit « == » (différent du signe « = » employé pour l'affectation).

Exemple de code dans la fonction « animation » :

```
x=x+A;
y=y+B;                                // incrementation ou decrementation
                                        // de y suivant la valeur de B

if(x>=R && x<=(W-R)){                 // bordure 1 ou 3
if(y <= R)B=1;                        // bordure haute
if(y >=(H-R))B=-1;                   // bordure basse
if(x>=pos_x && x<=pos_x+longueur && y==pos_y-R)B=-1; // test balle sur raquette
if( y> pos_y-R) alert("Fin de partie"); Pop-up "Fin de partie"
}

if(y>=R && y<=(H-R)){                 // bordure droite ou gauche
if(x <= R)A=getRandom(1,5);           // bordure gauche
if(x >=( W-R))A=-getRandom(1,5);     // bordure droite
}
```

4. Affichage du message fin de partie

La partie est terminée lorsque la balle dépasse la raquette : $y > \text{pos_y} - R$.
Une autre solution est de surveiller si la balle dépasse le cadre : $y > H - R$.

La fonction « alert » arrête le programme en affichant le message « Fin de partie » : `alert("Fin de partie");` appelée pop-up

Le code complet est donc

```
if (y > pos_y - R) alert("Fin de partie");
```