

BOÏNG BOÏNG, ÇA REBONDIT DE PARTOUT

Dans le cas de notre balle rebondissante, on appellera la variable `positionX`, pour mieux refléter ce à quoi elle sert : stocker la position de la balle suivant l'axe horizontal X. On en profite pour ajouter une variable `speedX` qui va servir à stocker la vitesse horizontale de la balle.

Ensuite, la balle va avancer d'elle même, pas besoin de **condition if** avec les boutons. On aurait donc quelque chose comme ça.

```
#include <Gamebuino-Meta.h>

int positionX = 32;
int speedX = 1;

void setup() {
  gb.begin();
}

void loop() {
  while (!gb.update());
  gb.display.clear();

  positionX = positionX + speedX;

  gb.display.print("positionX: ");
  gb.display.print(positionX);
  gb.display.print("\nspeedX: ");
  gb.display.print(speedX);

  gb.display.fillRect(positionX, 32, 4, 4);
}
```

Hé, mais la balle ne s'arrête pas ! Normal, on n'a mis aucune **condition**, elle continue donc d'avancer, même en dehors de l'écran. Comment est-ce qu'on pourrait faire? Voilà la logique à avoir, en pseudo-code.

Si la balle atteint le bord droit
Alors aller à gauche

Si la balle atteint le bord gauche
Alors aller à droite

Deux questions se posent alors...

- Comment faire pour que la balle aille à gauche ?
- Comment savoir quand on a atteint de bord de l'écran ?

Je vais vous donner la solution, mais vous devriez essayer d'y réfléchir un peu avant de regarder. Quelques indices : essayez de jouer avec la valeur de `speedX` pour faire varier la vitesse. Qu'est ce qui se passe avec une vitesse de 2, de 3 ? Une vitesse négative ?

Maintenant que vous avez trouvé le truc pour faire aller la balle à gauche (n'est-ce pas ?), vous n'avez plus qu'à détecter quand `positionX` atteint le bord de l'écran. Les coordonnées horizontales en X peuvent varier de 0 à `gb.display.width()`. Une valeur de 0 correspond au bord gauche de l'écran et une valeur de `gb.display.width()` correspond au bord droit.

Vous avez maintenant tous les éléments pour faire rebondir la balle de gauche à droite :)

Si vous bloquez, voilà un coup de pouce.

```
#include <Gamebuino-Meta.h>

int positionX = 32;
int speedX = 1;

void setup() {
  gb.begin();
}

void loop() {
  while (!gb.update());
  gb.display.clear();

  positionX = positionX + speedX;

  //si la balle atteint ou dépassé le bord gauche
  if(positionX < 0){
    //on part à droite
    speedX = 1;
  }

  //si la balle atteint ou dépassé le bord droit
  if(positionX > gb.display.width()){
    //on part à gauche
    speedX = -1;
  }

  gb.display.print("positionX: ");
  gb.display.print(positionX);
  gb.display.print("\nspeedX: ");
  gb.display.print(speedX);

  gb.display.fillRect(positionX, 32, 4, 4);
}
```

Si la balle ne bouge qu'à l'horizontale, ça ne va pas être très excitant comme partie de Pong. Pourquoi ne pas créer de nouvelles variables `positionY` et `speedY` pour permettre à la balle de bouger verticalement ? Il faudra aussi des conditions pour détecter le bord de l'écran, cette fois avec `gb.display.height()` qui donne la hauteur de l'écran.

Si vous bloquez, voici un exemple de solution :

```
#include <Gamebuino-Meta.h>

int positionX = 32;
int speedX = 2;
int positionY = 32;
int speedY = 1;
int ballSize = 8;

void setup() {
  gb.begin();
}

void loop() {
  while (!gb.update());
  gb.display.clear();

  //mise à jour de la position horizontale
  positionX = positionX + speedX;

  //si la balle atteint le bord gauche
  if(positionX < 0){
    //on part à droite
    speedX = 2;
  }

  //si la balle atteint le bord gauche
  if(positionX > gb.display.width() - ballSize){
    //on part à droite
    speedX = -2;
  }

  //mise à jour de la position verticale
  positionY = positionY + speedY;

  //si la balle atteint le bord haut
  if(positionY < 0){
    //on part en bas
    speedY = 1;
  }

  //si la balle atteint le bord bas
  if(positionY > gb.display.height() - ballSize){
    //on part en haut
    speedY = -1;
  }

  gb.display.setColor(BROWN);
  gb.display.print("positionX: ");
  gb.display.print(positionX);
  gb.display.print("\nspeedX: ");
  gb.display.print(speedX);

  gb.display.print("\npositionY: ");
  gb.display.print(positionY);
  gb.display.print("\nspeedY: ");
  gb.display.print(speedY);

  gb.display.setColor(WHITE);
  gb.display.fillRect(positionX, positionY, ballSize, ballSize);
}
```