

étape **2**

ARDUINO - PROCESSING

Utilisation d'un potentiomètre
pour ordonner un processus aléatoire

1 . Librairie Serial Call Response

Projet personnel

Elisa Raffy

1 Objectif du projet

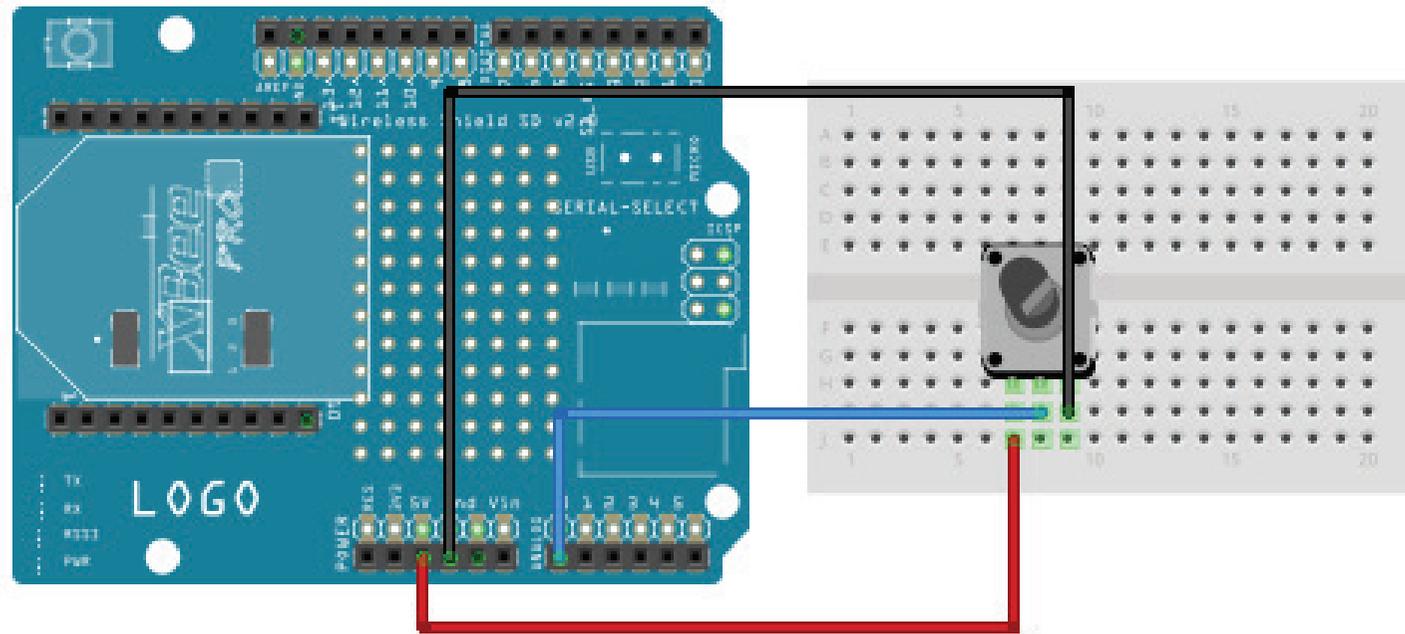
Faire varier un processus aléatoire de balles à l'aide d'une résistance variable (potentiomètre).

L'intensité de l'éclatement des balles se situe sur une abscisse «x». En tournant le potentiomètre de la gauche vers la droite, on obtient une destruction du chiffre «4». Cette destruction s'intensifie aussi en fonction de la rapidité du mouvement. Pour cela, il nous faut un montage arduino précis (voir ci-dessous), le sketch processing qui permet de faire varier une classe de balles, et la Librairie Serial-CallResponse (sur arduino et processing).

Librairie SerialCallResponse

-Elle permet de faire le lien entre processing et arduino.
(librairie disponible dans les exemples)

2 Montage



Outils :

- carte arduino + cable usb
- potentiomètre
- breadbord
- 3 fils conducteurs

3 *sur* ARDUINO

Étapes - déroulement

Ouvrir la librairie **SerialCallResponse** :
fichier > exemples > communication > SerialCallResponse.

Sur arduino, cette librairie permet (*vis- versa*) de détecter les signales de la carte et de les transmettre à l'ordinateur.

Ici, l'ouverture du **setup** signal la connection au port série.

L'ouverture du signal du port série commence a 9600 bps.

pinMode : défini la patte n°2 de la carte comme une sortie numérique.

Envoi d'un octet pour établir un contact avec la carte et avoir une réponse

```
void setup()
{
  Serial.begin(9600);
  while (!Serial) {

  }

  pinMode(2, INPUT);
  establishContact();
}
```

```
void loop()
{
```

Void loop est une fonction qui execute une boucle sans fin, permettant au programme de s'exécuter et de donner un signal. L'opération sera répétées une infinité de fois tant que la carte est sous tension.

```
if (Serial.available() > 0
```

— **if** : condition a executer si Serial.available est supérieur à 0

```
inByte = Serial.read();
```

— lecture de l'octet rentrant

```
firstSensor = analogRead(A0)/4;
```

```
delay(10);
```

— Lire la première entrée analogique, diviser par 4 pour faire la gamme 0-255:

— Demande à la carte d'attendre 10ms

4 *sur* ARDUINO

Étapes - déroulement

`void establishContact` est un parametage pour établir et attesté le contact entre carte et ordinateur.

Si le contact est établi, envoyer la lettre A.

```
void establishContact() {  
    while (Serial.available() <= 0) {  
        Serial.print('A');  
        delay(300);  
    }  
}
```

1 *sur* PROCESSING

Étapes - déroulement

Ouvrir la librairie **SerialCallResponse** :

sketch > import Library > Serial

Une fois la librairie ouverte, il faut remplacer la valeur dans le **mouseX**, par la valeur du potentiomètre : ici **xpose**.

1. Onglet Serial my port + Géométrie

Import de la librairie «Serial my port»

Décompte du nombre d'octets reçu.

Position de la balle de départ : en «x»

Signale reçu du micro contrôleur
«false» indique l'état du boolean, qui dans ce cas présent est «faux».

```
import processing.serial.*;

Serial myPort;

int serialCount = 0;
int xpos;
int inByte;
boolean firstContact = false;
```

// Le **Boolean** peut être soit vrai soit faux.

True si c'est vrai, et false si c'est faux.

Le boolean quand il est utilisé avec des valeurs permet de déterminer le débit d'un programme.

2 *sur* PROCESSING

Étapes - déroulement

2. Onglet Serial Event

La librairie Serial Event est appelée dès que les données sont détectés et disponibles de la carte à l'ordinateur.

La carte arduino envoie et récupère les données détectés

inByte est une variable qui lis les valeurs de ce qu'il y a sur le port usb

Dans arduino, envoyer un «A» si connecter/ou pas, si inByte est égale à «A», effacer le tampon du port série

Détéction des premiers contacts avec le microcontrolleur

«xpos» est ici la valeur du potentiomètre

Le signale détecte la lettre «A»

```
void serialEvent(Serial myPort) {
  int inByte = myPort.read();

  if (firstContact == false) {
    if (inByte == 'A') {
      myPort.clear();
      firstContact = true;
      myPort.write('A');
    }
  }
  else {

    xpos = inByte;

    println(xpos + "\t");
    myPort.write('A');

  }
}
```