

# **MANUEL DE MONTAGE DE LA MAQUETTE**

**CINÉTIQUE ET ARDUINO**

Le Bail Alix et Brosseau Marine MO3



## INTENTION:

Pour ce projet, nous avons souhaité suggérer la surprise chez le spectateur.

Afin de provoquer cette réaction nous avons décidé de se concentrer sur un mouvement des yeux et de la langue en s'inspirant de l'univers de Tex Avery et des réactions surjouées des personnages extrêmement expressifs et qui déjoue la loi de la physique.

L'idée étant que lorsque qu'une personne passe devant notre maquette, les yeux et la langue jaillissent vers elle.

Nous nous sommes intéressées au Yôkai. Ce sont des créatures surnaturelles qui peuplent l'imaginaire Japonais (figures fantasmagoriques). Le concept d'esprits farceurs et malicieux, qui s'amusent à jouer de mauvais tours aux humains, nous a beaucoup inspiré pour la suite. C'est donc cette idée que nous avons décidé de mettre en place à travers la mise en mouvement de la maquette.

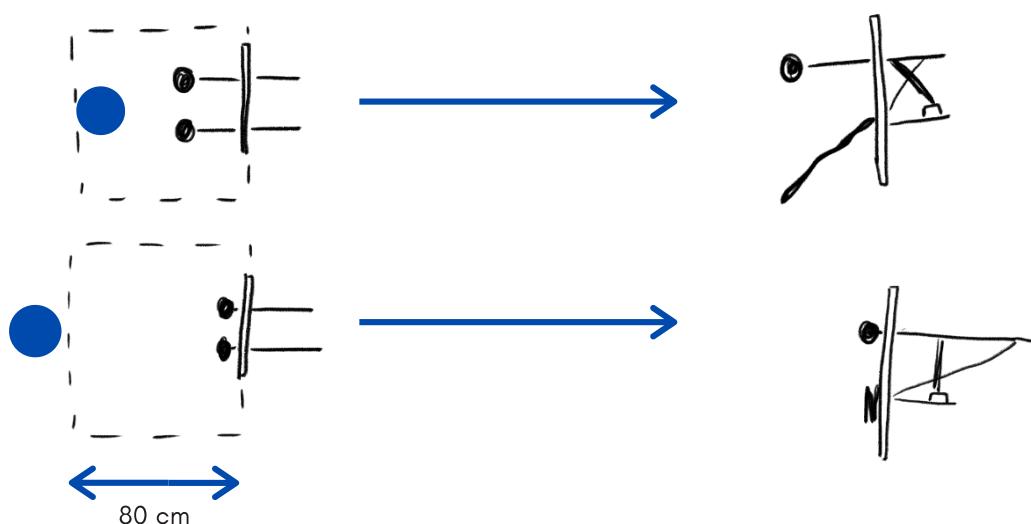
Nous avons pris le parti pour le graphisme de nous inventer un Yôkai, en s'inspirant des représentations des peintres de l'époque d'Edo.



Extraits du Hyakkai zukan (« rouleaux de dessins de cent yōkai ») achevé en 1737 durant l'époque Edo par l'artiste Sawaki Sūshi

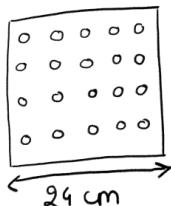
## SCÉNARIO UTILISATEUR:

Lorsque le spectateur est à une distance inférieure de 80cm de la maquette, les yeux et la langue jaillissent. Cette action surprend alors la personne. Puis lorsque le spectateur recule (distance supérieure à 80cm de la maquette), les yeux et la langue reviennent à leur place.

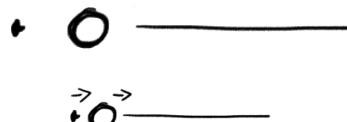


## LISTE DE MATÉRIEL

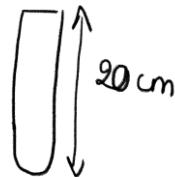
### POUR LE MÉCANISME:



1 planche en bois  
trouée de 24x24cm



2 punaises noires  
2 boules de polystyrène  
2 barres en bois de 0.25 mm de diamètre  
à assembler pour former un œil sur une tige



1 langue en papier  
de 5x20cm



Du fil transparent



Du fil



3 barres en bois de 0.5 mm de  
diamètre  
5 bâtonnets (plat)

### POUR LE FONCTIONNEMENT MÉCANIQUE:



1 carte Arduino



1 plaque d'essai



1 capteur de  
distance ultra-  
son-grove



1 servomoteur avec  
une hélice ronde



8 fils de branchements :  
- 3 rouges  
- 3 noirs  
- 2 de couleurs

1 visse

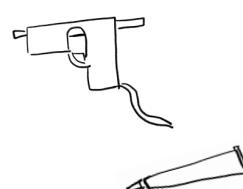
### POUR LE GRAPHISME:



Des pièces en bois  
découpées à la découpe  
laser



Du raphia



1 pistolet à colle  
De la colle liquide



Du papier crépon  
(1 rouge et 1  
blanc)

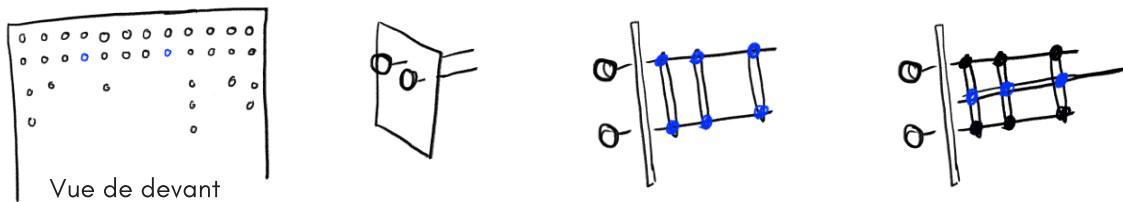


Du papier de  
soi noir

## CONCEPTION DU MÉCANISME

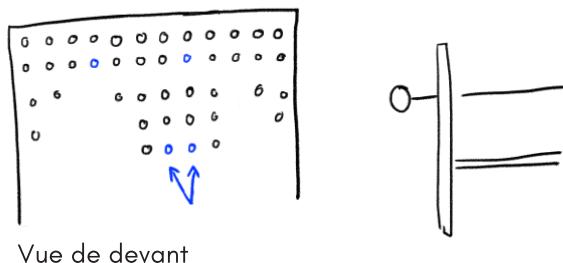
### ÉTAPE 1

Recouvrir la planche de bois de papier crépon blanc. Prendre les deux baguettes des yeux, y enfoncez les boules de polystyrène puis les punaises. Mettre une baguette dans un trou puis la deuxième dans un autre trou de la plaque à même hauteur. Relier ces 2 baguettes avec 3 bâtonnets. Ensuite placer une baguette au centre et la fixer aux bâtons.



### ÉTAPE 2

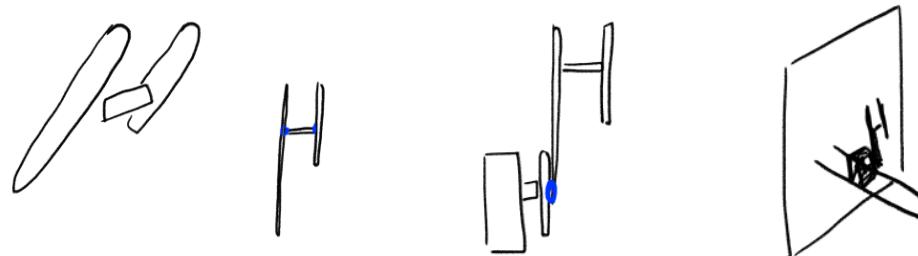
Prendre les 2 dernières baguettes et les enfoncez dans les trous en dessous des yeux. Ils serviront de supports pour poser le servomoteur.



Vue de devant

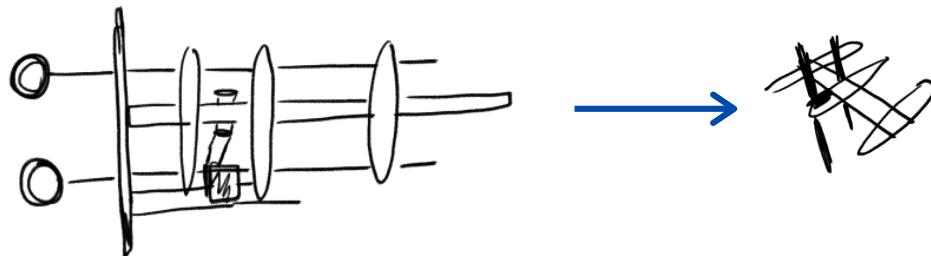
### ÉTAPE 3

Prendre les bâtonnets. En casser un en petit bout de 1 cm environ. Les fixer avec le pistolet à colle de manière à faire un H. Ce H sera alors vissé sur l'hélice du servomoteur qui sera lui-même fixé aux deux précédentes baguettes.



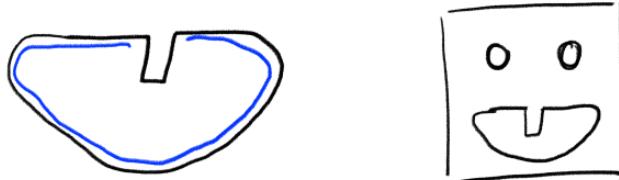
### ÉTAPE 4

Assembler le tout en plaçant la barre centrale dans le H et le H entre les 2 premiers bâtonnets.

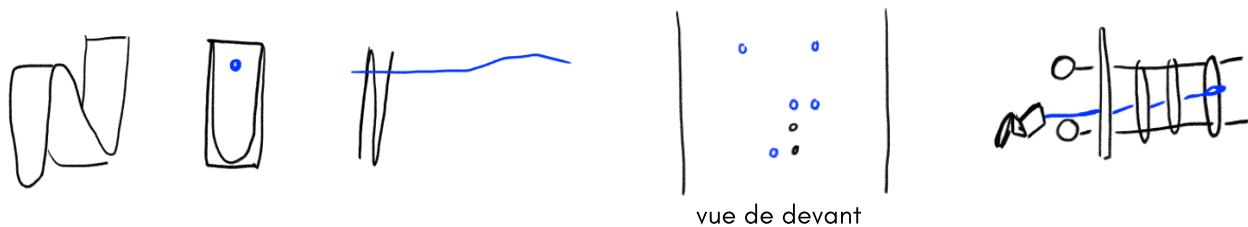


## ÉTAPE 5

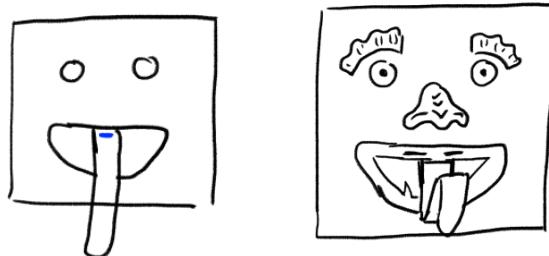
Découper la forme de la bouche dans le papier de soi noir. Faire une encoche assez large en haut au centre. Appliquer de la colle sur les bords puis coller.



Prendre la langue en papier et coller des côtés du papier crépon rouge. Plier la langue en 3 et faire un trou en haut. Y insérer un fil transparent de 20 cm environ. Puis faire passer ce fil dans le trou au milieu des deux yeux et relier ce fil au dernier bâton à glace.

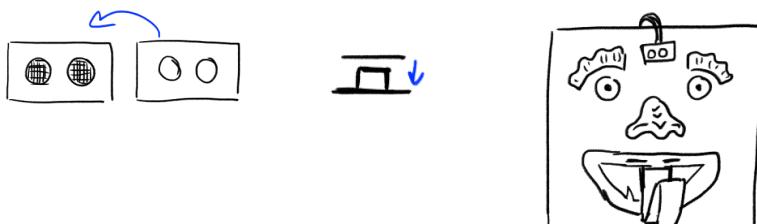


Coller le haut de la langue en haut du papier de soi noir. Coller la bouche et différents éléments en bois.



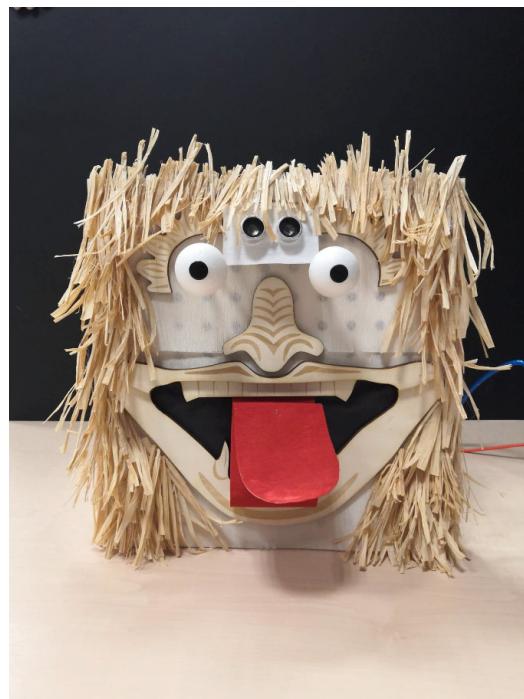
## ÉTAPE 6

Prendre le capteur de distance ultra-son-grove. Prendre du papier crépon blanc redessiner les trous puis l'enfoncer sur le capteur afin de le camoufler. Mettre le capteur entre les deux yeux par le dessus.



## ÉTAPE 7

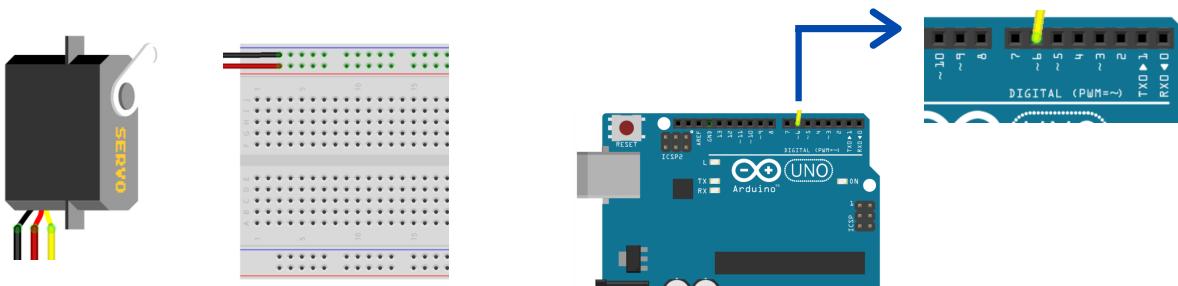
Découper des brins de raphia et en faire des petits fagots avec le pistolet à colle. Les fixer sur la maquette.



## ÉLABORATION DU CABLAGE

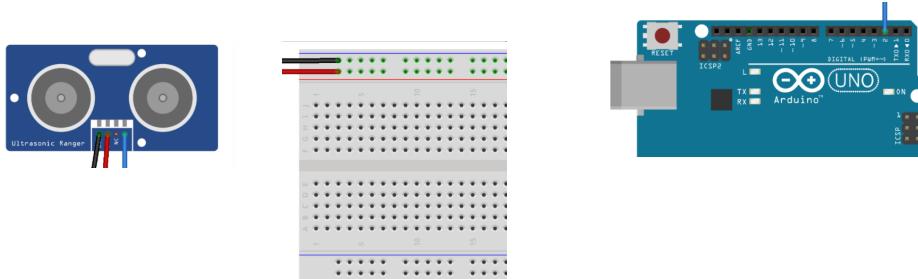
### ÉTAPE 1 : servomoteur

Brancher le servomoteur à la carte Arduino et à la plaque d'essai. Prendre un fil rouge pour l'alimentation. Le relier entre le capteur et la plaque d'essai. Le brancher sur le pôle positif. Prendre un fil noir pour la masse. Le relier entre le capteur et la plaque d'essai. Le brancher sur le pôle négatif. Prendre un fil de couleur. Le relier entre le servomoteur et la carte Arduino. Le brancher sur la sortie digitale "6".



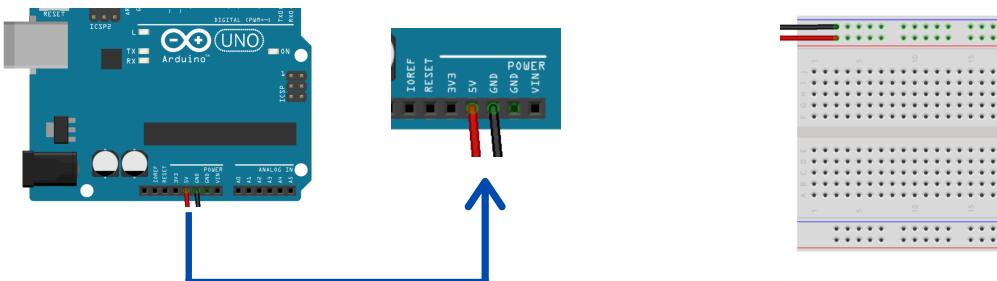
### ÉTAPE 2 : capteur de distance ultra-son-grove

Brancher le capteur à la carte Arduino et à la plaque d'essai. Prendre un fil rouge pour l'alimentation. Le relier entre le capteur et la plaque d'essai. Le brancher sur le pôle positif. Prendre un fil noir la masse. Le relier entre le capteur et la plaque d'essai. Le brancher sur le pôle négatif. Prendre un fil de couleur. Le relier entre le capteur et la carte Arduino. Le brancher sur la sortie digitale "9".

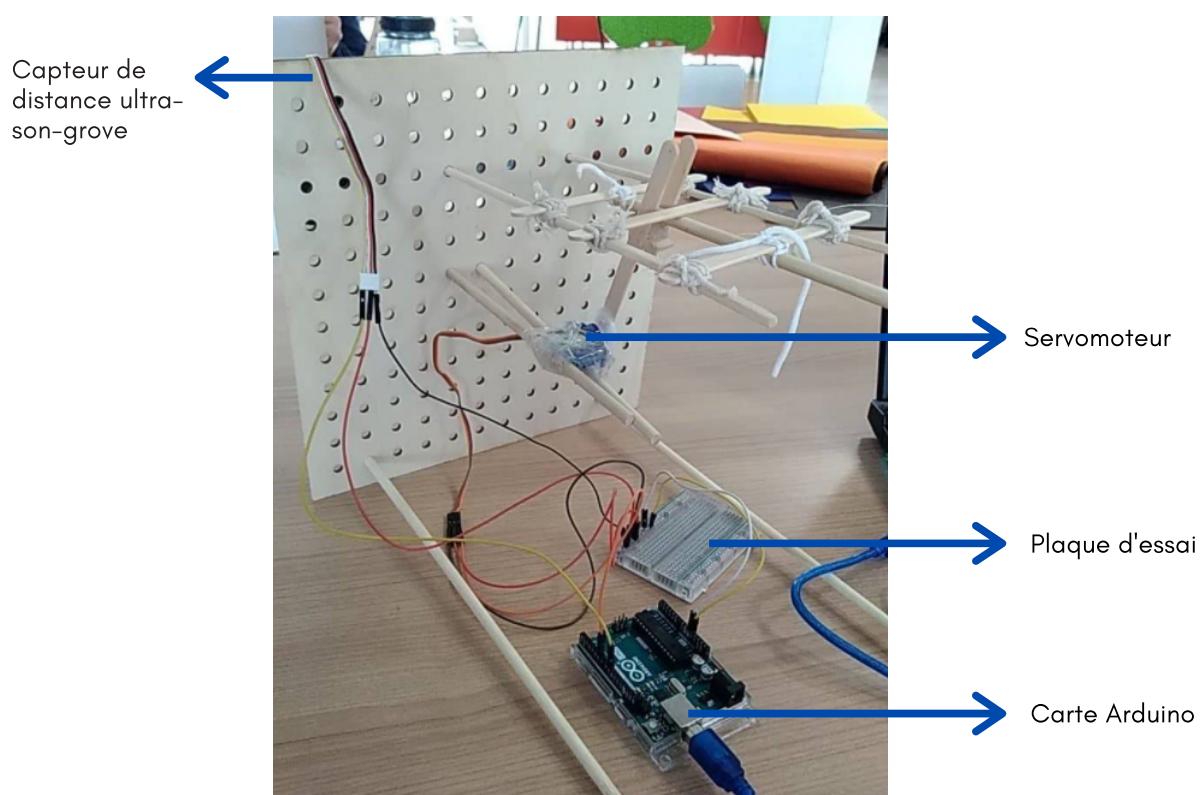


### ÉTAPE 3 : carte Arduino et plaque d'essai

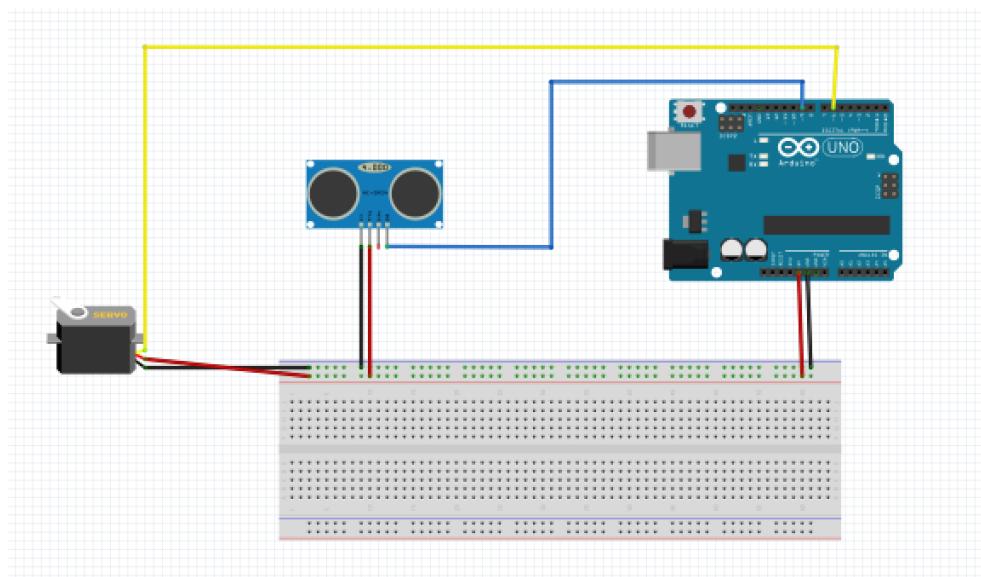
Brancher la carte Arduino à la plaque d'essai. Prendre un fil rouge pour l'alimentation. Le brancher d'un côté à la carte Arduino sur "5V" et de l'autre à la plaque d'essai sur le pôle positif. Prendre un fil noir pour la masse. Le brancher d'un côté à la carte Arduino sur "GND" et de l'autre à la plaque d'essai sur le pôle négatif.



## ÉLABORATION DU CABLAGE



## SCHÉMA FINAL



## EXPLICATION DU CODE

### ÉTAPE 1 : INTRODUCTION DES BIBLIOTHÈQUES ET OBJETS

Introduire le programme de la carte Arduino.

```
1 #include <Arduino.h>
```

Installer la bibliothèque Servo qui nous permettra de contrôler et de coder le servomoteur. Pour cela aller sur "Croquis" -> "Inclure une bibliothèque" -> "Gérer les bibliothèques" puis chercher et installer la bibliothèque. Sur la ligne de code introduire la bibliothèque Servo. Pour cela noter en dessous de "include <Arduino.h>" :

```
3 #include "ServoEasing.h"
```

Créer un objet qui correspond au servomoteur. Cela nous permettra de créer un lien entre le servomoteur et le code et ainsi pouvoir le manipuler par la suite.

```
5 #include <Servo.h>
```

Donner un petit nom au servomoteur. Ici nous l'appellerons "rota" comme "rotation". Cela permettra de nous repérer et de le retrouver plus facilement si jamais le code s'avère être long.

```
6 ServoEasing rota;
```

Installer la bibliothèque Grove Ultrasonic ranger qui nous permettra de contrôler et de coder le servomoteur. Pour cela aller sur "Croquis" -> "Inclure une bibliothèque" -> "Gérer les bibliothèques" puis chercher et installer la bibliothèque. Sur la ligne de code introduire la bibliothèque Servo.

```
8 #include "Ultrasonic.h"
```

Indiquer à notre bibliothèque que la broche "sig" du capteur est branchée sur la sortie digitale "9" de la carte Arduino.

```
9 Ultrasonic ultrasonic(9);
```

### ÉTAPE 2 : INSTALLATION

```
12 void setup() {
```

Ouvrir la connexion série. La valeur 9600 baud correspond au débit et donc à la vitesse.

```
13 Serial.begin(9600);
```

Indiquer que l'on souhaite utiliser la pin 6 comme sortie ("OUTPUT").

```
15 pinMode(6, OUTPUT);
```

Indiquer à notre bibliothèque que notre objet (servomoteur) est brancher sur la sortie digitale "6".

```
16 rota.attach(6);
```

## ÉTAPE 3 : CRÉER L'ACTION ET EN FAIRE UNE BOUCLE

```
20 void loop() {
```

### 1/ Servomoteur

Préciser à notre bibliothèque la vitesse d'action de notre servomoteur.

```
21 rota.setSpeed(100);
```

Ajouter un mouvement particulier à l'action du servomoteur (respecter les 12 principes d'animation). Ici nous précisons que le mouvement "bounce" se produira à la sortie ("OUT").

```
22 rota.setEasingType(EASE_BOUNCE_OUT);
```

### 1/ Capteur de distance ultra-son-grove

Nous précisons que pour le capteur de distance, nous allons mesurer la distance en cm et sur la deuxième ligne l'afficher.

```
24 long RangeInCentimeters = ultrasonic.MeasureInCentimeters();  
25 Serial.println(RangeInCentimeters);
```

Nous précisons que si la distance par rapport au capteur est inférieure à 80 cm, notre servomoteur réalisera l'action, c'est à dire la rotation souhaitée pour que les yeux sortent. (Normalement la valeur "0" indique la position "0" du servomoteur. C'est à dire sa position lorsque l'action n'a pas encore eu lieu. Mais suite à une erreur réalisée lors du montage, ici la valeur "0", à l'inverse, indique sa position lors de l'action.)

```
27 if (RangeInCentimeters < 80) {  
28     rota.easeTo(0);  
29 }
```

Nous précisons que dans le cas où la distance par rapport au capteur est autre qu'inférieure à 80cm, notre servomoteur reviendra à sa position initiale.

```
31 else {  
32     rota.easeTo(70);  
33 }
```

## ÉLABORATION DU CODAGE

Le\_Bail\_Alix\_Brosseau\_Marine\_code\_arduino | Arduino 1.8.13 (Windows Store 1.8.42.0)

Fichier Édition Croquis Outils Aide



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Le\_Bail\_Alix\_Brosseau\_Marine\_code\_arduino | Arduino 1.8.13 (Windows Store 1.8.42.0)
- File Menu:** Fichier, Édition, Croquis, Outils, Aide
- Toolbar:** Includes icons for Save, Undo, Redo, Open, Print, Upload, Download, and Televerser (Upload).
- Code Editor:** Displays the following C++ code for an Ultrasonic sensor project:

```
1 #include <Arduino.h>
2
3 #include "ServoEasing.h"
4
5 #include <Servo.h> //servomoteur
6 ServoEasing rota;
7
8 #include "Ultrasonic.h" // capteur 2
9 Ultrasonic ultrasonic(9);
10
11
12 void setup() {
13     Serial.begin(9600);
14
15     pinMode(6, OUTPUT); // servomoteur
16     rota.attach(6);
17 }
18
19
20 void loop() {
21     rota.setSpeed(100); // capteur 2
22     rota.setEasingType(EASE_BOUNCE_OUT);
23
24     long RangeInCentimeters = ultrasonic.MeasureInCentimeters();
25     Serial.println(RangeInCentimeters);
26
27     if (RangeInCentimeters < 80) {
28         rota.easeTo(0);
29     }
30
31     else {
32         rota.easeTo(70); // capteur 2
33     }
34 }
```