

Rapport 03/03/2022

Lors de cette séance, je me suis d'abord occupé de souder les piles entre elles afin de délivrer 2 * 9V en série, et de les brancher ensuite sur la breadboard afin d'alimenter le tout (Les piles marchent !)

Je me suis ensuite occupé de terminer le code du mode ultime :

-En repartant du code que j'avais initialement fait pour détecter les couleurs, j'ai ajouté des boucles for, et des contrôles des servomoteurs, pour que la caméra les détecte automatiquement et les cible.

```
#include <Servo.h>           //Servo
#include <Pixy2.h>           //Camera

Pixy2 pixy;
Servo turret_bas;
Servo turret_haut;

const int laser = 8;
String couleur = "bleu";

const String liste_coul[] = {"rouge", "orange", "jaune", "vert", "bleu", "violet", "marron"};

int pos_bas = 0;
int pos_haut = 135;

int x = 0;
int y = 135;
```

Je commence par définir des variables qui me serviront après, et initialiser certaines bibliothèques.

```
void setup() {
  Serial.begin(9600);
  turret_bas.attach(A0);
  turret_haut.attach(A1);
  pinMode(laser, OUTPUT);
  pixy.init();
}
```

Puis dans le void setup, je définie les différents pins, et j'initialise le module Pixy.

```

void loop()
{
    turret_haut.write(pos_haut);

    for (int j = 0; j < 150; j++)
    {
        turret_bas.write(j);
        turret_haut.write(135);
        pos_haut = 135;
        pos_bas++;
        delay(25);

        pixy.ccc.getBlocks();
    }
}

```

Puis, le void loop fonctionne en plusieurs parties. D'abord je mets le servomoteur à la position de base choisie arbitrairement, puis je fais balayer le servo du bas entre 0 et 150 degrés.

```

for (int i = 0; i < pixy.ccc.numBlocks; i++)
{
    Serial.println(liste_coul[pixy.ccc.blocks[i].m_signature - 1]);
    if (liste_coul[pixy.ccc.blocks[i].m_signature - 1] == couleur)
    {
        int x = pixy.ccc.blocks[i].m_x;
        Serial.println(x);
    }
}

```

Si une couleur voulue est détectée dans les blocs scannés, une variable garde en mémoire la position de ce bloc.

```

while (x > 165 or x < 155)
{
    if (x > 165)
    {
        pos_bas++;
        x -= 3;
        turret_bas.write(pos_bas);
        delay(15);
    }
    else
    {
        pos_bas--;
        x += 3;
        turret_bas.write(pos_bas);
        delay(15);
    }
}

```

Puis selon si le bloc se situe à droite ou à gauche du centre de la caméra (ici entre 155 et 165 pixels), le servomoteur va tourner pour se centrer.

```

int y = pixy.ccc.blocks[i].m_y;
while (y > 165 or y < 135)
{
    Serial.println(y);
    if (y < 135)
    {
        pos_haut ++;
        y += 3;
        turret_haut.write(pos_haut);
        delay(15);
    }
    else
    {
        pos_haut --;
        y -= 3;
        turret_haut.write(pos_haut);
        delay(15);
    }
}

```

Le même procédé est répété pour le servomoteur vertical, en prenant en compte que la caméra et le laser ne sont pas sur le même point sur l'axe vertical, il faut que la caméra se baisse un peu plus que le centre pour que le laser tire bien sur la cible.

```

        Serial.println("Tir");
        digitalWrite(laser, HIGH);
        delay(1000);
        digitalWrite(laser, LOW);
    }
}

for (int i=pos_bas; i>0; i--)
{
    turret_bas.write(i);
    pos_bas --;
    delay(15);
}

```

Et finalement, le laser s'allume pendant 1 seconde, et le servomoteur se replace à la position initiale.

Au début, le laser ne s'allumait pas à cause d'un blocage des pins 10 à 13 avec le module pixy, je l'ai donc déplacé sur le 8.

J'ai aussi rajouté du scotch pour tenir mieux la caméra qui risquait de se détacher avec les à-coups des servomoteurs, malgré la limitation faite grâce au code.