

# PROJET N°2 FLIPPER

BOUQUAIN Emilie - BOSSA Manon  
Groupe TPA  
2022-2023

## Table des matières




- PROJET N°2 FLIPPER
  - Présentation du sujet
  - Composants
  - Site des fournisseurs
  - Schéma
  - Affiche
  - Schéma logique initial
  - Schéma logique réel
  - Programme
  - Câblage
  - Maquette
  - Vidéo
  - Gantt
  - Améliorations
  - Complications
  - Webographie
  - Conseils
  - Remerciements






Modifier


## Présentation du sujet

- Notre projet consiste à créer un flipper utilisable pour tous. C'est à dire que toute les commandes qui doivent être activé à l'aide d'une main, par exemple le lancement de la balle, sera ,dans notre cas, activé par l'appuie d'un bouton qui lui peut-être enfoncer à l'aide d'un moignon. De même pour une personne à handicap mental l'appuie sur un bouton reste plus simple à comprendre.
- En France, 24% de la population est considéré comme handicapée.
- Jouer, c'est essayer de nouvelles choses, inventer, créer, résoudre des problèmes et explorer. Bien que les jeux soient, le plus souvent réserver aux enfants, il ne faut pas oublier que les adultes peuvent tirer les mêmes bienfaits du jeu que les enfants. Des recherches ont prouvé que le fait de s'amuser influençait notre santé physique et mentale. Exemples des bienfaits : Réduction du stress, amélioration de l'humeur et du bien-être émotionnel, stimulation du cerveau, stimulation de la curiosité, de la créativité et du goût d'apprendre, amélioration des relations interpersonnelles, accroissement de la productivité, dégourdissement de l'esprit et du corps, amélioration de la coordination, de la force et de la santé cardiovasculaire.
- Le jeu au travail réduit la fatigue, l'ennui, le stress et les risques d'épuisement professionnel. Il a aussi une incidence positive sur la satisfaction, le sentiment de compétence et la créativité.

## Composants

Numéro	Composant	Fournisseur 1	Prix	Fournisseur 2	Prix	Quantité	Caractéristiques
1	Arduino 	Go tronic	23€90	Conrad	24€16	1	Alimentation : 7 à 12 V / Modèle : Arduino UNO / 6 entrées analogiques
2	Module HC05 	Amazon	9€99	Go tronic	15€50	1	Module bluetooth
3	Servomoteur 	Go tronic	8,85	Lextronic	8,42	2	Alimentation : 4.8 à 6 Vcc / Dimension : 41*20*38 mm / Couple : - 3 kg.cm à 4.8 Vcc, - 3,2 kg.cm à 6 Vcc

4	Proximité à ultrason 	Amazon	6€90	Go tronic	4€50	2	Alimentation : 3,3 à 5 Vcc, Résolution : 1 cm
5	Leds  <b>X 50</b>	Amazon	3€89	Lextronic	5€90	1	Alimentation : - Rouge et jaune : 1,8 à 2 V, - Bleu, vert et blanc : 3 à 3,2 V / Taille : 5 mm, ronde / Quantité : 50
6	Buzzer 	RS France	5€24	Go tronic	3€60	1	Alimentation : 3 à 24 Vcc / Intensité : 95 dB
7	Afficheur lcd 	Amazon	6€99	Lextronic	12€	1	Alimentation : 3,3 et 5 V / Résolution : 16 caractères*2lignes / Taille : 80*36*12,5 mm
8	Capteur laser 	Go tronic	8€85	Amazon	6€79	1	

9	<p>Capteur de mouvement</p> 	Amazon	2€80	Farnell	3,72	3	Alimentation : 4,5 à 20 Vcc
10	<p>Capteur infrarouge</p> 	Amazon	11€79			1	
11	<p>Breadboard</p> 	Go tronic	4€50	RS France	5€22	1	408 connections possibles
12	<p>Ressort</p> 	Nous				2	
13	<p>Axe avec buttoir</p> 	Nous				1	
14	<p>Bille</p> 	Nous				1	

Nous n'avons finalement pas utilisé le capteur laser (8) car il ne fonctionnait pas et il n'y en avait plus en stock ; ni le capteur de mouvement (9) car sa zone de détection était trop grande. Le module HC05 (2) n'a pas servi car nous n'avons pas réussi à créer l'application.

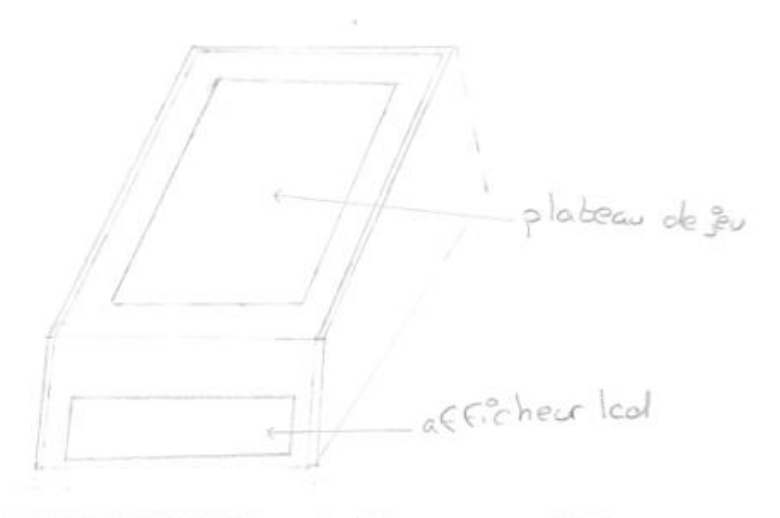
Modifier

## Site des fournisseurs

Conrad : <https://www.conrad.fr/ce/>  
Gotronic : <https://www.gotronic.fr/>  
Lextronic : <https://www.lextronic.fr/>  
RS : <https://fr.rs-online.com/web/>  
Amazon : <https://www.amazon.com/>  
Farnell : <https://fr.farnell.com/>

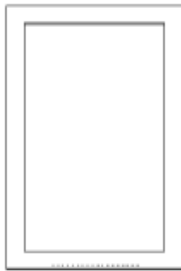
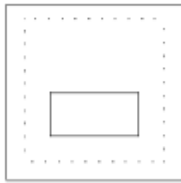
Modifier

## Schéma

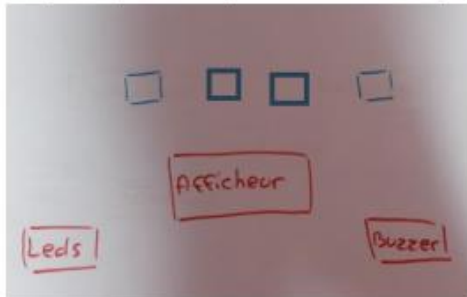


⊞ obstacle  
○ gain de points

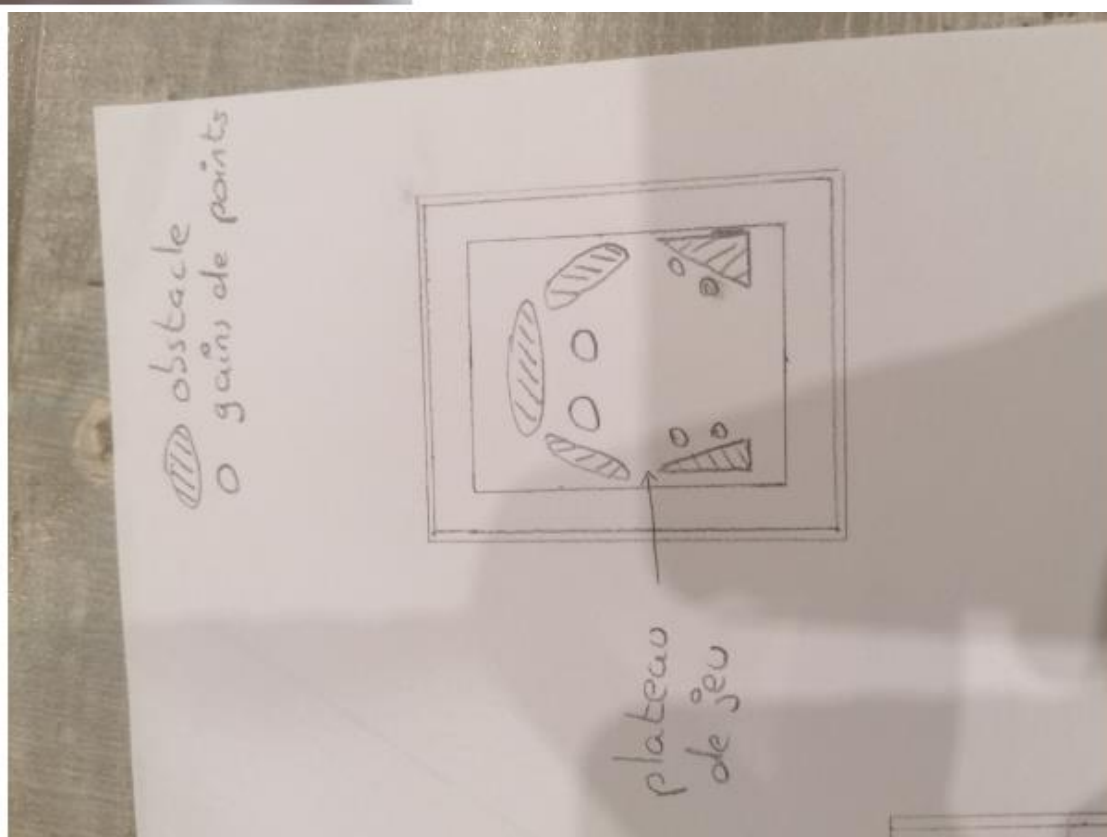
plateau  
de jeu

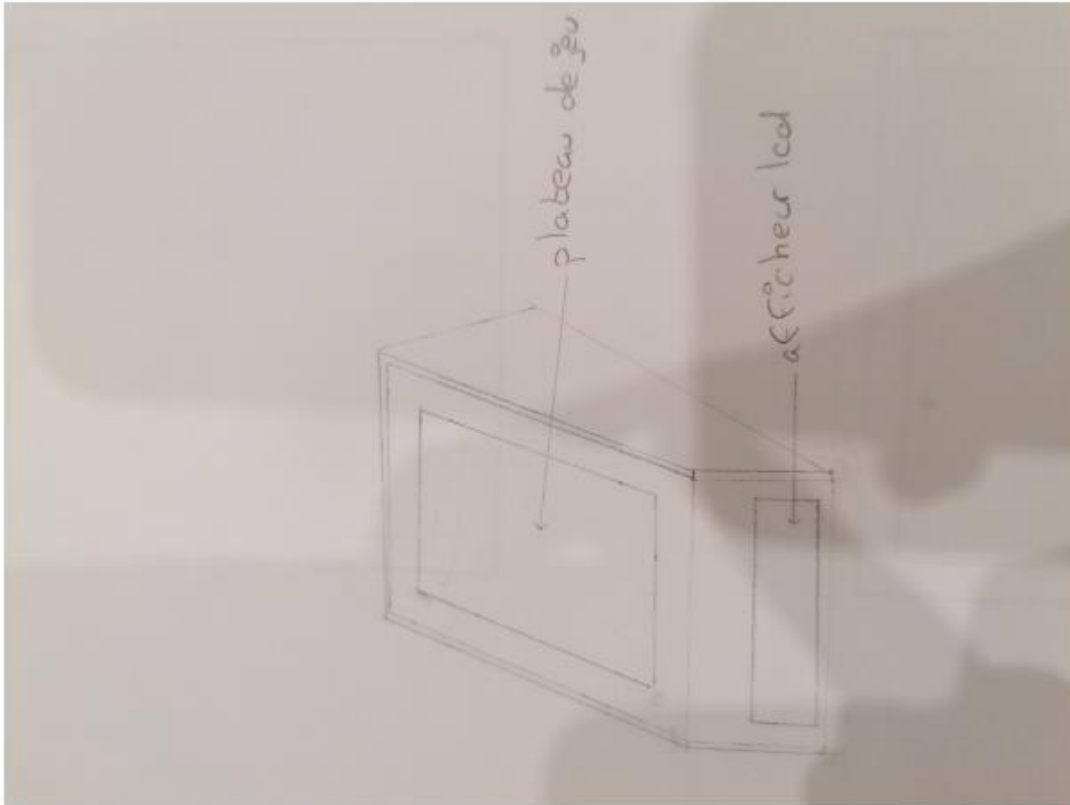


Ce qui est en pointillé n'est pas visible sur la vue en question. Ce que l'on verra ou à quoi nous aurons accès.



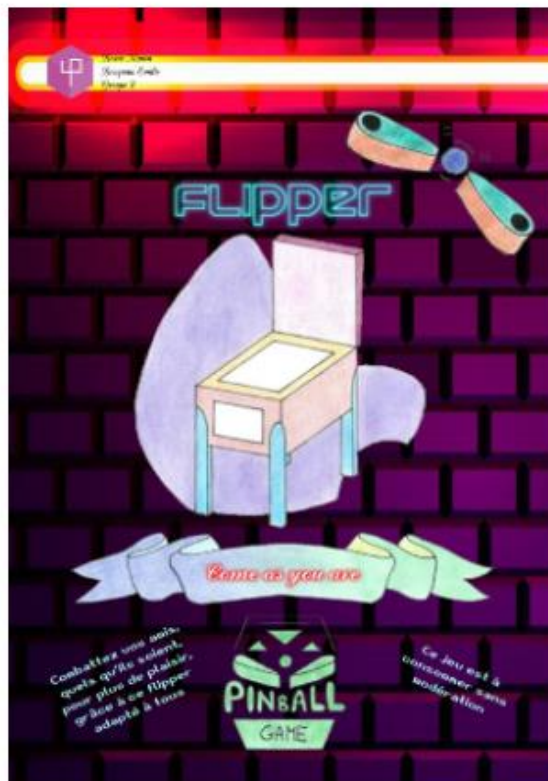
Ce qui sera à l'intérieur de la structure.





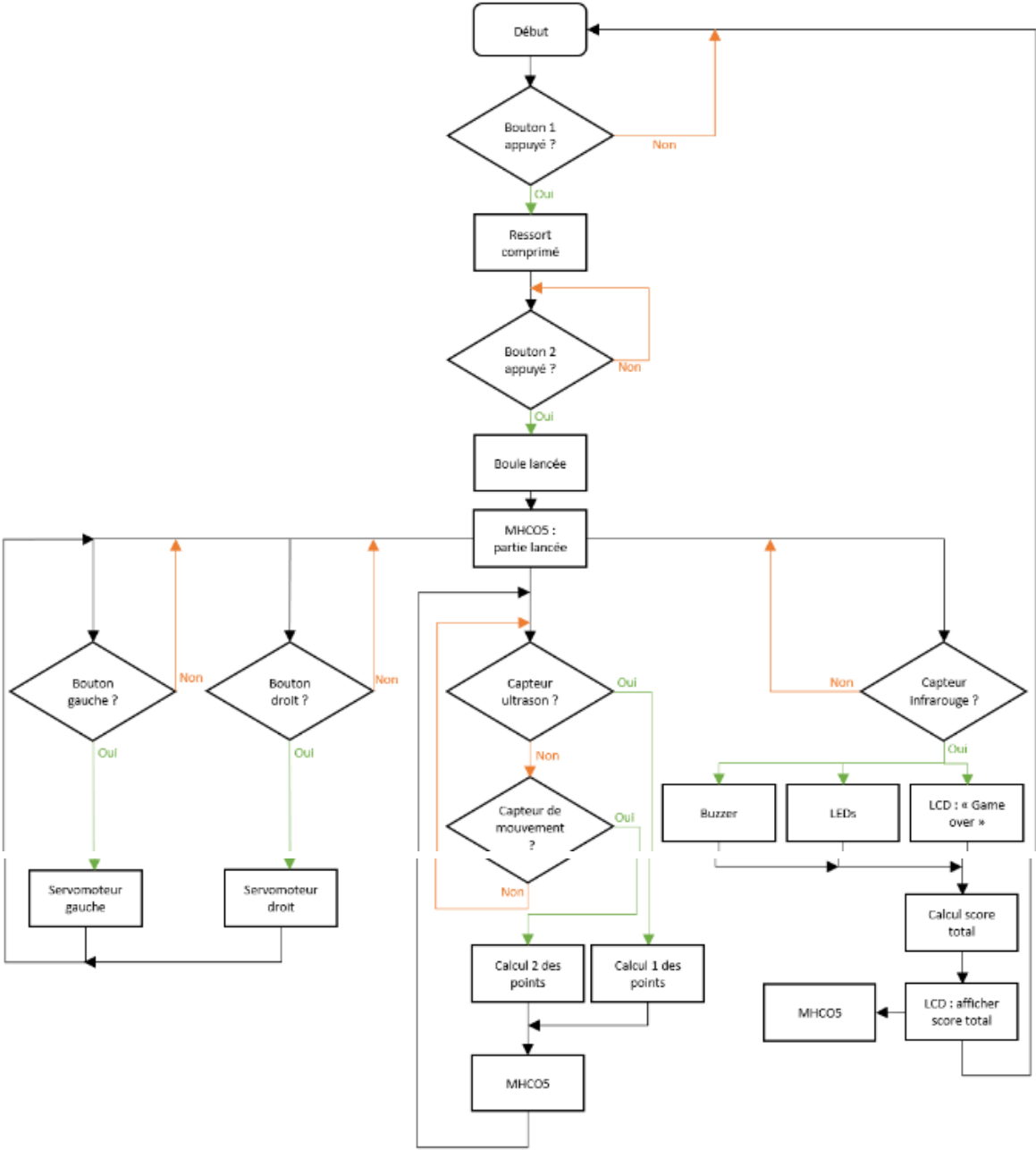
Affiche

74

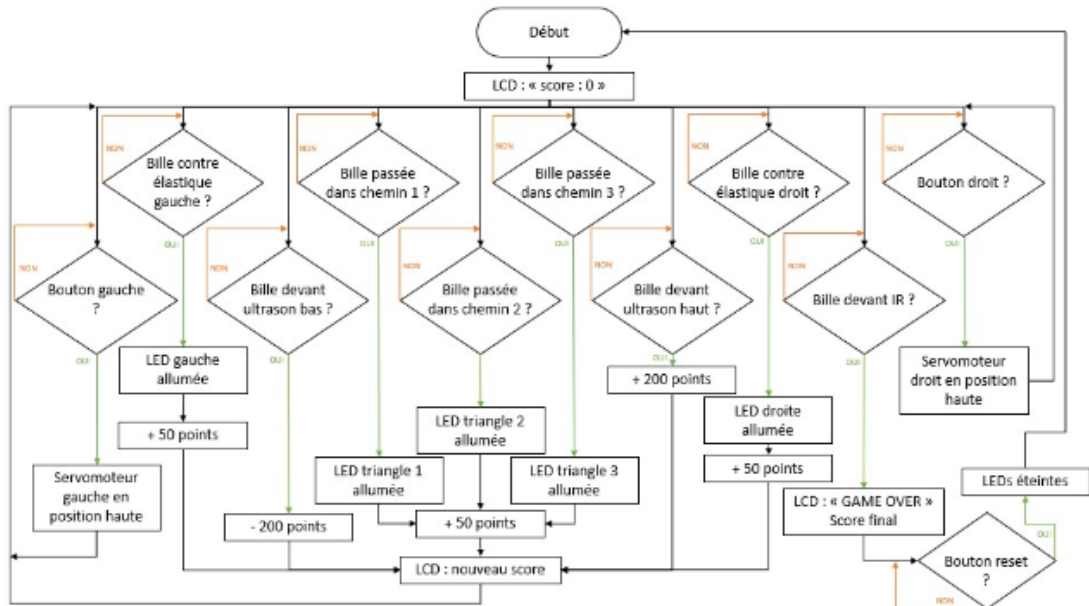


75

# Schéma logique initial



## Schéma logique réel



## Programme

```
//broches des capteurs ultrasons
const byte TRIGGER_US_haut = 4;
const byte ECHO_US_haut = 5;
const byte TRIGGER_US2 = 6;
const byte ECHO_US2 = 7;

//vitesse du son dans l'air en mm/us
const float SOUND_SPEED = 340.0 / 1000;

int BP_LED_gauche = 0;
int BP_LED_droite = 0;
int BP_LED_triangle1 = 1;
int BP_LED_triangle2 = 1;
int BP_LED_triangle3 = 1;
int BP_Reset = 0;
int IR = 0;
int GAME_OVER = 0;

//mémoire des boutons pour détecter seulement les fronts montants ou descendants
int M_BP_LED_gauche = 0;
int M_BP_LED_droite = 0;
int M_BP_LED_triangle1 = 1;
int M_BP_LED_triangle2 = 1;
int M_BP_LED_triangle3 = 1;
int M_BP_Reset = 0;

//pin des LEDs
int LED_gauche = 13;
int LED_droite = 9;
int LED_triangle1 = 10;
int LED_triangle2 = 11;
int LED_triangle3 = 12;

//score en début de partie
int score = 0;
```

```

//pin des servomoteurs qui permettent le mouvement des batteurs
int BP_servo_droit= 32 ;
int BP_servo_gauche= 34;

//buzzer
int buzzer = 40;
int i = 0;
int noteduration = 1000 / 8;
int pause = noteduration * 0.2;

//bibliothèques pour l'afficheur LCD
#include "Wire.h"
#include "LiquidCrystal_I2C.h"
LiquidCrystal_I2C LCD(0x27, 16, 2); // définit le type d'écran LCD 16 x 2

//servomoteurs + bibliothèque
#include "Servo.h"
Servo servo_gauche;
Servo servo_droit;

void setup() {

//capteur ultrason en haut du plateau qui donne des points
pinMode(TRIGGER_US_haut, OUTPUT); // TRIGGER est une sortie
digitalWrite(TRIGGER_US_haut, LOW); // La broche TRIGGER doit être à LOW au repos
pinMode(ECHO_US_haut, INPUT); // ECHO est une entrée

//capteur ultrason en bas du plateau qui donne une pénalité
pinMode(TRIGGER_US2, OUTPUT); // TRIGGER est une sortie
digitalWrite(TRIGGER_US2, LOW); // La broche TRIGGER doit être à LOW au repos
pinMode(ECHO_US2, INPUT); // ECHO est une entrée

// pins LEDs = sorties
pinMode(13, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);

```

```
// pins boutons = entrées
pinMode(22, INPUT);
pinMode(24, INPUT);
pinMode(26, INPUT);
pinMode(28, INPUT);
pinMode(36, INPUT);

pinMode(IR, INPUT); // pin capteur infrarouge = entrée

pinMode(40, OUTPUT); // pin buzzer = sortie

// pins boutons pour activer les servomoteurs = entrées
pinMode(BP_servo_gauche, INPUT);
pinMode(BP_servo_droit, INPUT);

// initialisation de l'afficheur
LCD.init();
LCD.backlight();

// pins des servomoteurs
servo_gauche.attach(2);
servo_droit.attach(3);

// mélodie buzzer
tone(buzzer, 500, 150);
delay (201);
tone(buzzer, 800, 300);
delay (201);
tone(buzzer, 500, 150);
delay (201);
tone(buzzer, 800, 900);
}
```

```
void loop() {

  BP_LED_gauche = digitalRead(22);
  BP_LED_droite = digitalRead(24);
  BP_LED_triangle1 = digitalRead(26);
  BP_LED_triangle2 = digitalRead(28);
  BP_LED_triangle3 = digitalRead(30);
  BP_Reset = digitalRead(36);
  IR = digitalRead(38);
  GAME_OVER = digitalRead(42);

  // fin de partie si bouton game over déclanché ou IR détecte la bille
  // bouton game over pas prévu mais ajouté car capteur IR pas bien adapté
  // capteur IR==LOW quand il détecte la bille et IR==HIGH quand il ne détecte pas
  if (GAME_OVER==HIGH||IR==LOW)
  {
    // mélodie game over
    for (i = 25; i < 35; i++)
    {
      tone(buzzer, 20 * i, noteduration);
      delay(pause);
    }
    for (i = 35; i >= 25; i--) {
      tone(buzzer, 20 * i, noteduration);
      delay(pause);
    }

    delay(500);
    tone(buzzer, 500, 200);
    delay(200);
    tone(buzzer, 1200, 200);
    delay(200);
    tone(buzzer, 300, 200);
    delay(200);
    tone(buzzer, 1000, 200);
    delay(200);
    tone(buzzer, 400, 200);
    delay(200);
    tone(buzzer, 1100, 200);
    delay(200);
  }
}
```

```

// afficher "game over" et le score final sur l'afficheur LCD
LCD.clear();
LCD.setCursor(4, 0);
LCD.print("GAME OVER");
LCD.setCursor(3, 1);
LCD.print("SCORE :");
LCD.print(score);
while(digitalRead(36)==LOW)
{
  // reste tant que bouton reset pas appuyé
}
}

// capteurs ultrasons
// Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche TRIGGER
digitalWrite(TRIGGER_US_haut, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_US_haut, LOW);
// Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe)
long measure_haut = pulseIn(ECHO_US_haut, HIGH);
//Calcul la distance à partir du temps mesuré
float distance_haut_mm = measure_haut / 2.0 * SOUND_SPEED;

digitalWrite(TRIGGER_US2, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_US2, LOW);
long measure2 = pulseIn(ECHO_US2, HIGH);
float distance2_mm = measure2 / 2.0 * SOUND_SPEED;

// afficher les distances mesurées sur le moniteur série
/* Serial.print(F("Distance haut:"));
Serial.print(distance_haut_mm);
Serial.print(F("mm"));
Serial.print(F("      Distance2:"));
Serial.print(distance2_mm);
Serial.println(F("mm"));*/

// gain de point si la bille est détectée par le capteur ultrason en haut du plateau
if(distance_haut_mm<30||distance_haut_mm>51)
{
  score=score+200; // + 200 points
  //mélodie buzzer
  tone(buzzer, 500, 100);
  delay (101);
  tone(buzzer, 1000, 100);
  delay(300);
}
else
{
  score=score;
}

// pénalité si la bille est détectée par le capteur ultrason en bas du plateau
if(distance2_mm>42)
{
  score=score-200; // - 200 points
  // mélodie buzzer
  if (score<0) score=0;
  tone(buzzer, 800, 100);
  delay (101);
  tone(buzzer, 500, 100);
  delay (101);
  tone(buzzer, 200, 100);
}
else
{
  score=score;
}
}

```

```
// points avec l'élastique (marche comme un bouton) à gauche
// détecte seulement les fronts montants pour compter une seule fois les points à chaque fois que la bille tape l'
if (BP_LED_gauche != M_BP_LED_gauche)
{
  M_BP_LED_gauche = BP_LED_gauche;

  if (BP_LED_gauche == 1)
  {
    score = score + 50; // + 50 points
    digitalWrite(13,HIGH); // allume LED
    // mélodie buzzer
    tone(buzzer, 500, 100);
    delay (101);
    tone(buzzer, 800, 100);
  }
}

// points avec l'élastique (marche comme un bouton) à droite
// détecte seulement les fronts montants pour compter une seule fois les points à chaque fois que la bille tape l'
if (BP_LED_droite != M_BP_LED_droite)
{
  M_BP_LED_droite = BP_LED_droite;

  if (BP_LED_droite == 1)
  {
    score = score + 50; // + 50 points
    digitalWrite(LED_droite,HIGH); // allume LED
    // mélodie buzzer
    tone(buzzer, 500, 100);
    delay (101);
    tone(buzzer, 800, 100);
  }
}
}
```

```

// points lorsque la bille passe dans le premier chemin (marche à l'inverse d'un bouton)
// détecte seulement les fronts descendants pour compter une seule fois les points à chaque fois que la bille pass
if (BP_LED_triangle1 != M_BP_LED_triangle1)
{
  M_BP_LED_triangle1 = BP_LED_triangle1;

  if (BP_LED_triangle1 == 0)
  {
    score = score + 50; // + 50 points
    digitalWrite(LED_triangle1,HIGH); // allume LED
    // mélodie buzzer
    tone(buzzer, 500, 50);
    delay (51);
    tone(buzzer, 800, 50);
    delay (51);
    tone(buzzer, 650, 50);
  }
}

// points lorsque la bille passe dans le deuxième chemin (marche à l'inverse d'un bouton)
// détecte seulement les fronts descendants pour compter une seule fois les points à chaque fois que la bille pass
if (BP_LED_triangle2 != M_BP_LED_triangle2)
{
  M_BP_LED_triangle2 = BP_LED_triangle2;

  if (BP_LED_triangle2 == 0)
  {
    score = score + 50; // + 50 points
    digitalWrite(LED_triangle2,HIGH); // allume LED
    // mélodie buzzer
    tone(buzzer, 500, 50);
    delay (51);
    tone(buzzer, 800, 50);
    delay (51);
    tone(buzzer, 650, 50);
  }
}

// points lorsque la bille passe dans le troisième chemin (marche à l'inverse d'un bouton)
// détecte seulement les fronts descendants pour compter une seule fois les points à chaque fois que la bille pass

```

```

// points lorsque la bille passe dans le troisième chemin (marche à l'inverse d'un bouton)
// détecte seulement les fronts descendants pour compter une seule fois les points à chaque fois que la bille pass
if (BP_LED_triangle3 != M_BP_LED_triangle3)
{
  M_BP_LED_triangle3 = BP_LED_triangle3;

  if (BP_LED_triangle3 == 0)
  {
    score = score + 50; // + 50 points
    digitalWrite(LED_triangle3,HIGH); // allume LED
    // mélodie buzzer
    tone(buzzer, 500, 50);
    delay (51);
    tone(buzzer, 800, 50);
    delay (51);
    tone(buzzer, 650, 50);
  }
}

// tout remettre à 0 pour relancer une partie à l'appui du bouton reset (bouton du milieu)
if (BP_Reset == HIGH)
{
  // éteint toutes les LEDs
  digitalWrite(LED_gauche, LOW);
  digitalWrite(LED_droite, LOW);
  digitalWrite(LED_triangle1, LOW);
  digitalWrite(LED_triangle2, LOW);
  digitalWrite(LED_triangle3, LOW);
  // remet le score à 0
  score = 0;
  // efface game over et ancien score de l'afficheur LCD
  LCD.clear();
  LCD.setCursor(5, 0);
  LCD.print("score :");
  LCD.setCursor(5, 1);
  LCD.print(score);
  // mélodie buzzer
  tone(buzzer, 500, 150);
  delay (201);
  tone(buzzer, 800, 300);
}

```

```

    delay (201);
    tone(buzzer, 500, 150);
    delay (201);
    tone(buzzer, 800, 900);
}

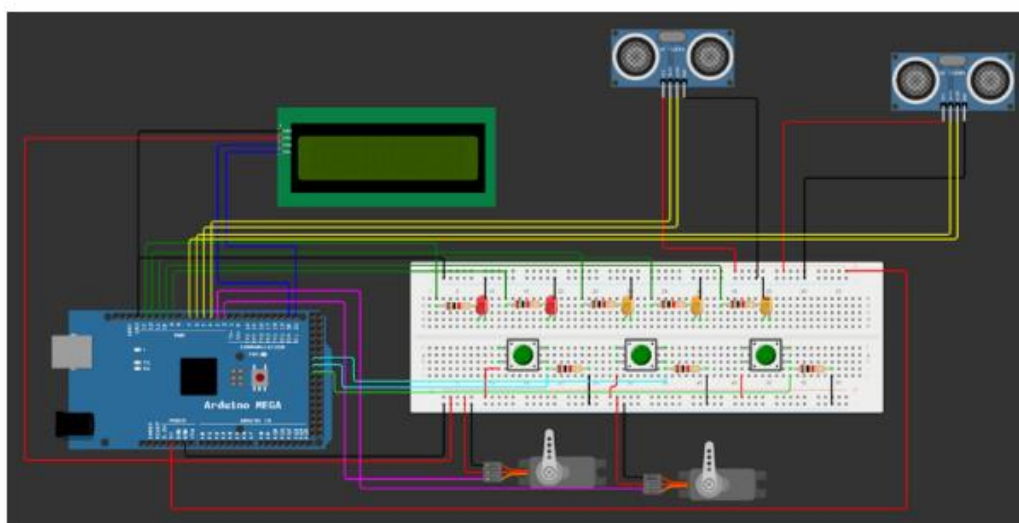
// servomoteurs
// à l'appui du bouton gauche le batteur gauche est en haut
if (digitalRead(BP_servo_gauche) == HIGH)
{
    servo_gauche.write(55);
}
// position initiale du batteur gauche
if (digitalRead(BP_servo_gauche) == LOW)
{
    servo_gauche.write(90);
}
// à l'appui du bouton droit le batteur droit est en haut
if (digitalRead(BP_servo_droit) == HIGH)
{
    servo_droit.write(38);
}
// position initiale du batteur droit
if (digitalRead(BP_servo_droit) == LOW)
{
    servo_droit.write(8);
}

// affichage du score sur l'afficheur LCD tout au long de la partie
LCD.setCursor(5, 0);
LCD.print("score :");
LCD.setCursor(5, 1);
LCD.print(score);
}

```

## Câblage

Voies	Capteurs/Actionneurs
13	LED Gauche
12	LED triangle 3
11	LED triangle 2
10	LED triangle 1
9	LED Droite
20	SDA écran
21	SCL écran
2	servo 1
3	servo 2
4	Trigger ultrason 1
5	Echo ultrason 1
6	Trigger ultrason 2
7	Echo ultrason 2
22	BP LED gauche
24	BP LED droit
26	BP LED triangle 1
28	BP LED triangle 2
30	BP LED triangle 3
34	BP servo gauche
32	BP servo droit
36	Reset
38	IR
40	Buzzer
42	BP game over

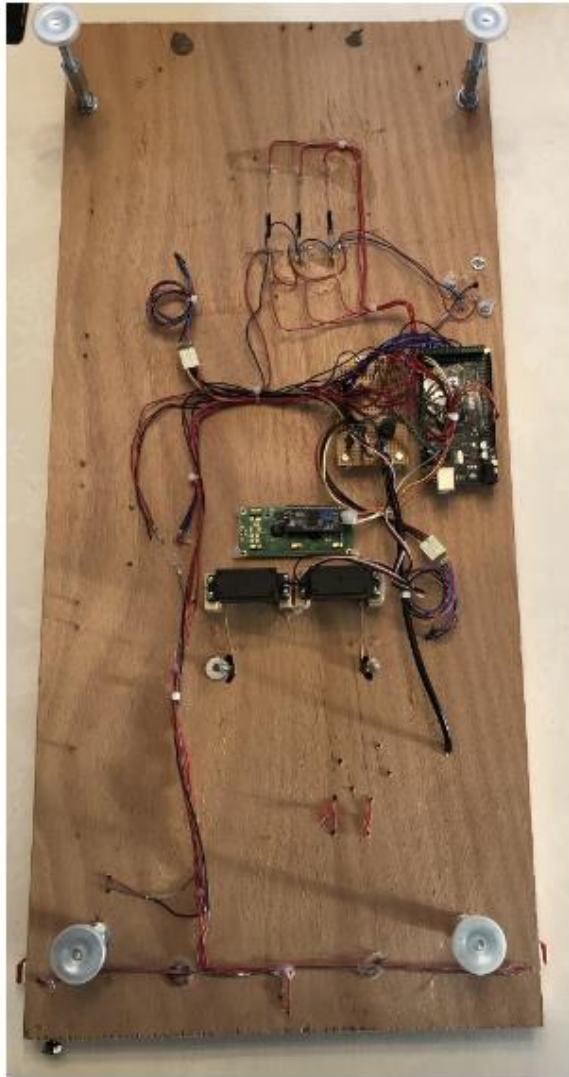


Sur ce schéma les LEDs jaunes représentent les LEDs triangulaires. Les cibles fabriquées par nous mêmes (BP LED gauche, BP LED droit, BP LED triangle 1, BP LED triangle 2, BP LED triangle 3, BP game over), le capteur infrarouge et le buzzer ne sont pas représentés.

[Modifier](#)

## Maquette





 [Diaporama maquette PowerPoint](#)

 [Diaporama maquette PDF](#)

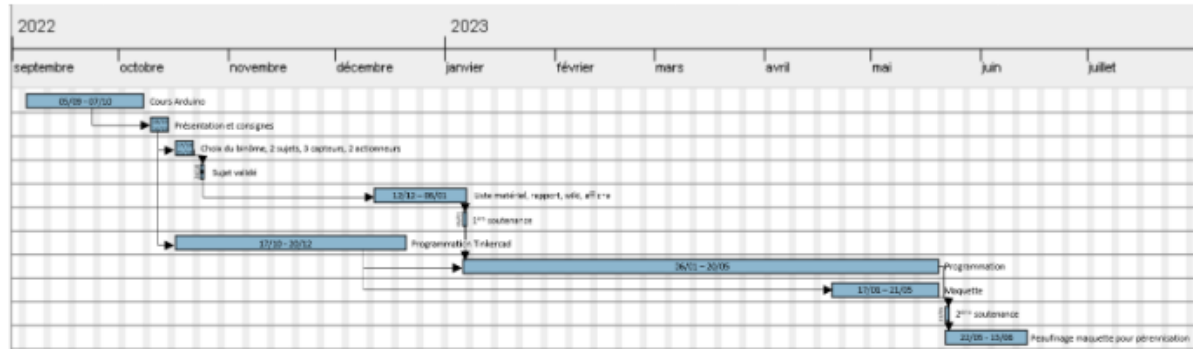
Modifier

## Vidéo

 <https://youtube.com/shorts/v8hLid0b-mk>

Modifier

## Gantt



Modifier

## Améliorations

Nous avons pensé à créer un obstacle à l'aide d'un servomoteur qui se lève à partir d'un certain nombre de points, ainsi qu'un système de récompenses. Nous souhaitions créer une application grâce à un module bluetooth mais nous n'avons pas réussi à aller au bout. Nous avons réussi à communiquer de l'application à la carte mais pas de la carte à l'application. Le but de l'application était de compter le score, afficher le détail du score, retenir le meilleur score...

Modifier

## Complications

Les servomoteurs consomment beaucoup de courant, la pile s'use très vite. Nous avons réussi à adapter le lanceur avec un servomoteur afin de le rendre automatique pour que toute personne, même avec un handicap, puisse utiliser le flipper. Mais, suite à la découverte de la surconsommation des servomoteurs, nous avons décidé de fabriquer un lanceur mécanique. Certains programmes fonctionnaient et ont cessé de fonctionner d'un moment à un autre, nous avons dû recommencer plusieurs programmes suite à des dysfonctionnements. Certains capteurs ne fonctionnaient pas du tout, la difficulté a été de trouver si le problème venait du capteur ou du programme. Certains capteurs choisis au début du projet n'étaient pas adaptés à nos besoins, nous avons alors changé certains capteurs.

Modifier

## Webographie

 [Tinkercad](#)

 [Guide Arduino](#)

 [Vidéo afficheur LCD](#)

 [Gantt](#)

Modifier

## Conseils

Il ne faut pas baisser les bras même si c'est dur. Mais, il faut surtout s'amuser et réussir à aller au bout. Il est essentiel de maîtriser et répartir son temps intelligemment.

Modifier

## Remerciements

Nous tenons à remercier HOSATTE Didier, CARILLAT Myriam, MAKOUDI Younes, VUILLEMENT Yannick, BOUVEROT Véronique, ainsi que l'institut universitaire de technologie pour nous avoir permis de réaliser ce projet. Ainsi que de s'améliorer dans certaines compétences, comme le travail d'équipe, la programmation, le respect d'un cahier des charges; trois savoir-faire très importants pour la poursuite de nos études et le monde professionnel.

Modifier