

Jeu de tir laser

Projet Arduino PEIP2



Lannier Clarisse et Morillon Camille

03/2020

1. Table des matières

2. Description de projet	3
Présentation	3
Objectifs	3
Cahier des charges.....	3
3. Planning	4
4. Développement du projet	4
Matériel	4
Difficultés et solutions	7
5. Conclusion et perspective d'amélioration	8
6. Annexe.....	8

2. Description de projet

Présentation

Notre projet est un jeu de tir laser composé d'un support avec trois cibles, activées à l'aide de l'application Bluetooth Electronics. Les cibles sont visées avec un pistolet laser. Une cible touchée correspond à un point rajouté au score qui s'affiche sur l'écran situé au milieu du support.

Objectifs

Premièrement, nous voulions un support pour les cibles dans lequel on puisse ranger tous les composants c'est-à-dire la carte Arduino Genio Uno, les plaques pour brancher les fils et les différents modules. Mais nous voulions également que ce soit esthétique et que l'on voit du premier coup d'œil que c'est un jeu. Pour le design des cibles nous ne voulions pas juste faire des cercles mais quelque chose de plus original. Notre première idée était de trouver un trio de méchant connus. Nous n'avons pas trouvé de trio mais nous avons remarqué qu'il existe 3 contes connus avec des grands méchants loups. Donc nous avons décidé de les représenter sur nos cibles. Nous voulions aussi faire le pistolet nous même en l'imprimant en 3d. L'objectif étant de pouvoir ouvrir ce pistolet pour pouvoir accéder à ce qui se trouve à l'intérieur.

Cahier des charges

Les différentes contraintes auquel nous avons dû faire face sont que le projet soit transportable, doit être fait en huit séances, un coût peu élevé et une connexion radio fréquence.

Les caractéristiques du jeu de tir laser sont qu'il y a des cibles qui se lèvent commandées par une application mobile à l'aide du Bluetooth. Un laser doit toucher les cibles pour qu'elles se baissent. D'autres fonctionnalités peuvent être rajoutées, par exemple un score.

3. Planning

Nous n'avons pas respecté notre planning initial. En effet, nous avons prévu beaucoup trop de séances pour le développement du code car nous ne connaissions pas les composants que nous allions utiliser. De plus, il s'est avéré que ces composants étaient similaires à ceux vus en cours. A contrario nous ne pensions pas mettre autant de temps pour effectuer la conception du pistolet et de la boîte. Par ailleurs, nous n'avons pas respecté l'ordre que nous avons fixé pour certaines parties car certaines sont dépendantes des autres. Par exemple, les photorésistances avaient besoin des servos-moteurs et nous ne l'avions pas prévu. Ou alors nous n'avions pas pensé que nous aurions besoin de faire certaines tâches comme souder des fils puisqu'ils n'étaient pas assez long.

Finalement nous avons pris suffisamment d'avance sur le code pour pouvoir rajouter le score qui s'affiche sur l'écran LCD.

1	Planning initial	Décembre		Janvier		Février			Mars			
2		09/12/2019	16/12/2019	06/01/2020	13/01/2020	03/02/2020	10/02/2020	17/02/2020	02/03/2020	09/03/2020	16/03/2020	25/03/2020
3	montage avec planche a essai				ORAL							ORAL
4	code: faire lever les cibles				ORAL							ORAL
5	code: configurer le laser				ORAL							ORAL
6	code: contrôler avec bluetooth				ORAL							ORAL
7	code: ajouter le bouton				ORAL							ORAL
8	code: capter le laser				ORAL							ORAL
9	assemblage: construire support, cible,...				ORAL							ORAL
10	assemblage: préparation du pistolet				ORAL							ORAL
11	assemblage: ajouter les cablages cibles et pistolets				ORAL							ORAL
12												
13	Planning réel	Décembre		Janvier		Février			Mars			
14		09/12/2019	16/12/2019	06/01/2020	13/01/2020	03/02/2020	10/02/2020	17/02/2020	02/03/2020	09/03/2020	16/03/2020	25/03/2020
15	montage avec planche a essai				ORAL							ORAL
16	code: faire lever les cibles				ORAL							ORAL
17	code: configurer le laser				ORAL							ORAL
18	code: contrôler avec bluetooth				ORAL							ORAL
19	code: ajouter le bouton				ORAL							ORAL
20	code: capter le laser				ORAL							ORAL
21	assemblage: construire support, cible,...				ORAL							ORAL
22	assemblage: préparation du pistolet				ORAL							ORAL
23	assemblage: ajouter les cablages cibles et pistolets				ORAL							ORAL
24	écran/score											

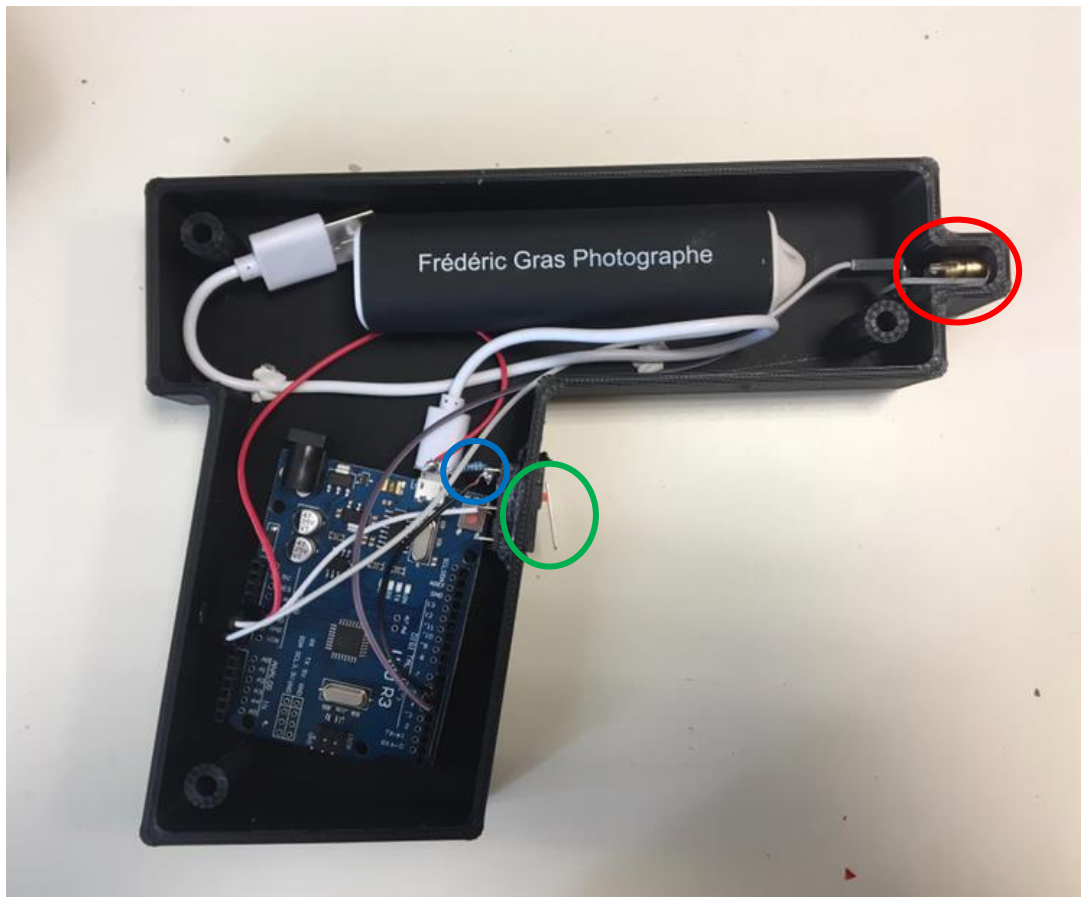
4. Développement du projet

Matériel

Pour le pistolet laser, le matériel utilisé est :

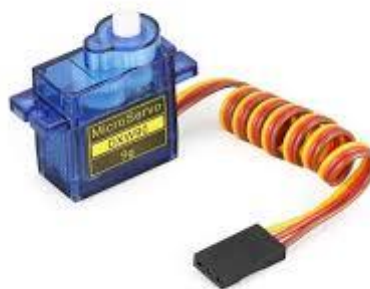
- Un pistolet en plastique imprimé en 3D
- Une batterie externe

- Une carte Arduino Genio Uno
- Un bouton poussoir sous forme de gâchette
- Un module laser
- Une résistance 10k Ω
- Des fils



Pour le support avec les cibles :

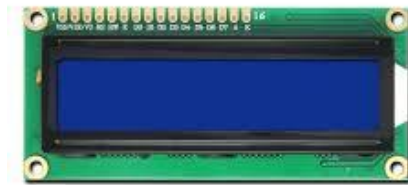
- Trois servo-moteurs



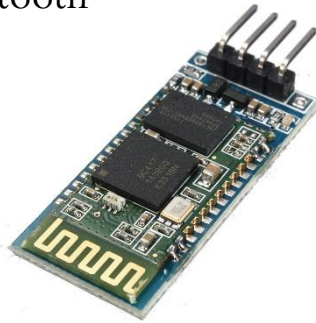
- Trois photorésistances



- Trois résistances de $10k\Omega$
- Une carte Arduino Genio Uno
- Deux plaques pour brancher les fils
- Fils
- Une alimentation
- Un écran LCD



- Le module Bluetooth



- Un condensateur de $100\mu F$
- Un condensateur de $1000\mu F$
- Trois bâtons de glaces
- Trois cibles
- Du bois
- Des vis



Difficultés et solutions

Pour le pistolet la partie la plus compliquée a été de faire le design du pistolet sur le logiciel Autodesk Inventor. En effet, le pistolet a dû être recommencé plusieurs fois car l'épaisseur de celui-ci était trop fine, les dimensions étaient dans un premier temps trop grandes puis trop petites, ou car les marges étaient trop ou pas assez grandes pour compenser l'imprécision de l'imprimante 3D. De plus, après l'impression la partie qui devait s'emboîter ne coïncidait pas. Donc cette pièce a été refaite en rajoutant un bord pour que l'emboîtement soit soutenu de façon plus homogène.

Pour le support, un des problèmes que nous avons rencontrés est pour les photorésistances car la luminosité de la pièce était forte et cela faussait les calculs pour déterminer la valeur de la photorésistance touchée par le laser. Pour le score, le problème était qu'il comptait à l'infini puisque la boucle continuait tant que le laser touchait la photorésistance. Ainsi il a fallu rajouter une condition supplémentaire pour qu'il ne re-rentre pas dans le if. Pour l'écran, le problème était que l'écran fonctionnait par intermittence car il y avait une trop grosse chute de tension due aux trop grands nombres de modules. La solution a été de rajouter un condensateur. Nous avons eu le même problème pour les servos-moteurs d'où une alimentation supplémentaire et pour le Bluetooth d'où un deuxième condensateur. Le dernier problème rencontré a été entre les servos-moteurs et le Bluetooth : leurs librairies ne coïncidaient pas. Ainsi, il a fallu faire marcher le Bluetooth sans librairie en le branchant sur l'entrée 0 et la sortie 1.

5. Conclusion et perspective d'amélioration

Finalement nous avons tout de même réussi à faire tous ce que nous voulions. Nous avons également eu le temps de rajouter un module. Nous n'aurions jamais pensé avoir le temps de rajouter le score et de faire notre propre pistolet (nous voulions utiliser un nerf).

Mais il y a quand même des points à améliorer comme le pistolet puisque les parties s'emboitent mal. De plus nous pourrions rajouter un clapet à la boîte pour cacher et protéger les fils et les composants. Pour ce qui est du jeu nous pourrions le rendre plus autonome avec un bouton pour l'allumer et l'éteindre, et ajouter un temps pour chaque partie de jeu.

6. Annexe

```
pistolet$  
const int laser=6;  
const int bouton=7;  
int val=0;  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(laser,OUTPUT);           //on configure le laser en sortie  
  pinMode(bouton,INPUT);           //on configure le bouton en entrée  
  digitalWrite(laser,HIGH);        //on configure le laser en mode éteind  
}  
  
void loop() {  
  val=digitalRead(bouton);         //val prend les valeurs du bouton (HIGH ou LOW)  
  if (val==HIGH) {                 // si on appuie sur le bouton  
    digitalWrite(laser,LOW);       //on allume le laser  
  }  
  else {                            //sinon  
    digitalWrite(laser,HIGH);      //il reste éteind  
  }  
}
```



```

//cibles
const int cible0=A2;
const int cible1=A1;
const int cible2=A0;

int valeur0=0;
int valeur1=1;
int valeur2=2;

//bluetooth
char Data;

//servos
#include <Servo.h>
Servo servoG;
Servo servoM;
Servo servoD;

//ecran
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,9,10,11,12);

int compteur=-1;
int avant=90;
int apres=180;

void setup() {
  //bluetooth
  Serial.begin(9600);

  //cible
  pinMode(cible0, INPUT);
  pinMode(cible1, INPUT);
  pinMode(cible2, INPUT);

  //ecran
  lcd.begin(16,2);
  analogWrite(4,25);
  lcd.setCursor(8,0);

  //servo
  servoG.attach(6);
  servoM.attach(7);
  servoD.attach(8);
  servoG.write(90); //première position à 90°
  servoM.write(90);
  servoD.write(90);
}

```

```

void loop() {
  if(Serial.available()){
    Data=Serial.read();
    if (Data=='G'){
      servoG.write(180);
      apres=180;}
    if (Data=='M'){
      servoM.write(180);
      apres=180;}
    if (Data=='D'){
      servoD.write(180);
      apres=180;}
  }
  valeur0=analogRead(cible0);
  valeur1=analogRead(cible1);
  valeur2=analogRead(cible2);

  if (valeur0>900) {
    servoG.write(90);
    avant=90;}

  if (valeur1>900) {
    servoM.write(90);
    avant=90;}

  if (valeur2>900) {
    servoD.write(90);
    avant=90;}

  //affichage score
  if (avant==90){
    if (apres==180){
      compteur=compteur+1;}
    avant=180;
    apres=90;}

  lcd.setCursor(3,0);
  lcd.print("Score :");
  lcd.setCursor(11,0);
  lcd.print(compteur);
  Serial.println(compteur);
}

```