

Bentolux - BentoGhoost

Jeu d'adresse de type ciblirie électronique avec 3 fantômes à abattre au 3ème étage d'une Bentolux.

 Difficulté Moyen

 Durée 2 jour(s)

 Catégories Électronique, Bien-être & Santé, Jeux & Loisirs

 Coût 40 EUR (€)

Sommaire

Introduction

Video d'introduction

Étape 1 - Fabrication de la BentoBox

Fabrication 3ème étage

Étape 2 - Fabrication du plancher

Étape 3 - Fabrication du plancher (suite 1)

Étape 4 - Fabrication du plancher (suite 2)

Étape 5 - Fabrication du plancher (suite 3) porte-cible

Étape 6 - Fabrication du plancher (suite 4) remonte-cible

Étape 7 - Fabrication du plancher (suite 5) Les 3 microrupteurs

Étape 8 - Fabrication du plancher (suite et fin) rampe d'appui des cibles

Étape 9 - Fabrication en 3D du bouton potar

Étape 10 - Branchement électrique Arduino

Étape 11 - BOOOooUUUUUUuuuuu

Étape 12 - Un peu de fun avant le code ;-)

Étape 13 - Programme Arduino

Commentaires

Introduction

Ce tuto concerne la fabrication du troisième étage d'une Bentolux dans le cadre de la formation FabNumAura dispensé par l'EMSE (école des Mines de Saint-Etienne).

La réalisation de ce projet nous a permis de mettre en oeuvre les notions apprises à distance dans les MOOC de l'IMT Atlantique mais aussi en présentiel au FabLab de l'EMSE et le FabLab OpenFactory du quartier créatif Manufacture-Plaine Achille de Saint-Etienne.

funMOOC S'initier à la fabrication numérique - Programmer un objet Arduino -Imprimer en 3D

OpenFactory SITE du FabLab

GENESE

Au commencement, Dieu créa.... (non j'déconne)

Avant de commencer la formation, je m'étais fabriqué un stand de tir pour airsoft dans mon vide sanitaire avec un système me permettant de relever les cibles tombées à l'aide d'une corde d'un peu moins de 10 mètres.

Dès que nous avons abordé Arduino dans la formation je me suis dit.... Bon, mon système à corde fonctionne mais ce serait beaucoup plus fun de remonter automatiquement les cibles et pourquoi pas de se créer en plus des séquences de jeu différentes.

Cette envie est restée dans un coin de ma tête jusqu'au jour où nous devons réfléchir au projet "fil rouge" caractérisé par la création du troisième étages de notre Bentolux.

Nous devons créer des binômes pour la réalisation de cette étage libre...

Lors d'une pause café avec mes camarades de formation, je leur partage l'idée d'un troisième étage "stand de tir". Renaud me dit, si tu veux on le fait ensemble.

A ce moment là de la formation, je ne connaissais pas encore tout le monde et je ne savais pas que Renaud est un adepte de GN (jeu de rôle grandeur nature).

Pour ces parties de Shadowrun, il utilise des Nerf qu'il customise (entre autres accessoires) avec sa team.

Autant dire que la perspective de dégommer des cibles au Nerf n'était pas pour lui déplaire.

C'est ainsi que commença la créative et rocambolesque collaboration de deux quadras adultes à la chevelure fantomatique.

----- REPARTITION DES TACHES

Afin d'optimiser le temps qui nous était imparti (;-)), nous nous sommes répartis les différentes tâches ainsi:

Renaud

- Création du "gros oeuvre" sur Inkscape (box entourant les deux premiers étages de la Bentolux).
- Création des fantômes sur Inkscape (cibles+ceux en plexi des faces de la box).
- Découpe laser de la box, du deuxième étage et des cibles fantômes en contreplaqué.
- Découpe laser des fantômes en plexi vert incrusté sur les faces avant et latérales de la box.
- Assemblage et collage de la box et du deuxième étage de la Bentolux.
- Collage des fantômes en plexi dans les trous des faces de la box.
- Rédaction de toutes les étapes de la doc du Wikifab.

Mayak

- Création d'une maquette pour se représenter le mécanisme des cibles avec le système de relevage.
- Création sur Inkscape des pièces constitutives au mécanisme des cibles et celui du remonte-cible actionné par le servomoteur.
- Assemblage, collage, perçage et ajustement de toutes les pièces avec les microrupteurs et servomoteur.
- Création dans fusion 360 du bouton du potar (imprimé, installé, mais que nous n'utiliserons finalement pas dans notre version de base actuelle)
- Montage et câblage des composants électroniques sur la box et l'arduino.
- prise de vues (photos et vidéos) pour illustrer la doc du Wikifab.

Pour ce qui est du code nous y avons travaillé ensemble. Pour cette version de base, nous avons dû revoir à la baisse nos ambitions par manque de temps, mais nous comptons faire évoluer cette box que ce soit en terme de séquences/modes de jeu ou en nombre de modules additionnels de cibles.

----- MATOS & OUTILS

- Panneau de CP peuplier 3mm (plus épais pour la plateforme serait mieux)
 - Panneau de Plexiglass vert
 - 1 tige fileté de 6mm
 - des écrous et rondelles de 6mm

 - Scie à métaux
 - Pince, serre-joint
 - Equerre,règle, crayon...patience et minutie
 - Perceuse à colonne (ou perceuse...)
 - Fer à souder
 - Clé plate de 6mm (deux c'est mieux ou avec une pince à bec pour serrer les écrous entre eux)
 - Colle à bois
 - Colle à chaud (pistolet)
 - Colle forte Super Glue (pour les aimants)

 - 1 Arduino Uno
 - 1 breadboard
 - des fils électriques (beaucoup)
 - 5 leds
 - 5 résistances 220 kΩ
 - 1 servomoteur
 - 3 microrupteurs
 - 1 potar (sera utilisé dans la V2)
-

----- NOS PETITES GALERES (génératrices d'astuces et de partages)

Evidement nous avons dû résoudre deux trois petits soucis...

-la taille de notre box étant relativement importante, le CP de peuplier de 3mm est un peu trop fin et de fait la stabilité du plancher où repose le mécanisme des cibles n'est pas parfaitement plane.

-il a fallu lester les cibles avec un boulon afin qu'elles actionnent correctement les microrupteurs lors de leur chute.

-Nous avons préféré utiliser des aimants plutôt que les band "scratch" initiales pour maintenir plus sûrement les cibles à la verticale.

-Le potar ne rentrait pas dans le trou du bouton imprimé en PLA (c'était ma première impression), quelques secondes au mini décapeur thermique ont suffi pour régler le soucis.

J'en oublie sûrement plein mais du coup j'en profite pour remercier toutes les personnes du FabLab qui nous ont apporté leurs lumières (ou simplement indiqué l'emplacement d'outils) dans les phases obscures de notre projet.

Maxime, Gael, Thibaud, (les)Michaël... et Hubert évidemment qui a su nous faire redescendre sur terre sans jamais nous couper les ailes ;-)

MERCI à vous tous :-)

Matériaux

Outils

- 📄 Bentolux_-_BentoGhoost_BentoGhost_volume1.svg
- 📄 Bentolux_-_BentoGhoost_BentoGhost_fantomes.svg
- 📄 Bentolux_-_BentoGhoost_BentoGhost_cibles_et_servo.svg
- 📄 Bentolux_-_BentoGhoost_V_1_OK_BentoGhost.ino.ino

Étape 1 - Fabrication de la BentoBox

Fabrication 3ème étage

1. Fabrication à la découpeuse laser.

- Plan à télécharger : BentoGhost_volume1.svg (clic droit / enregistrer le fichier sous)

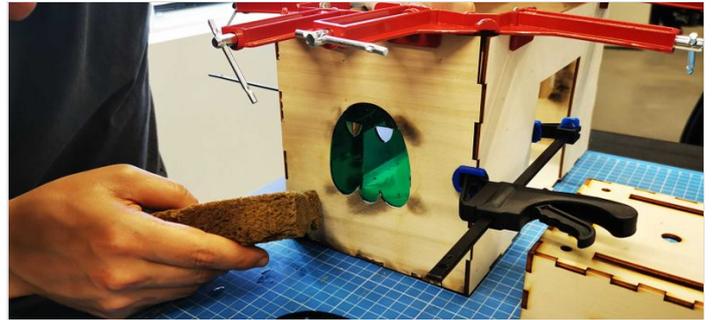
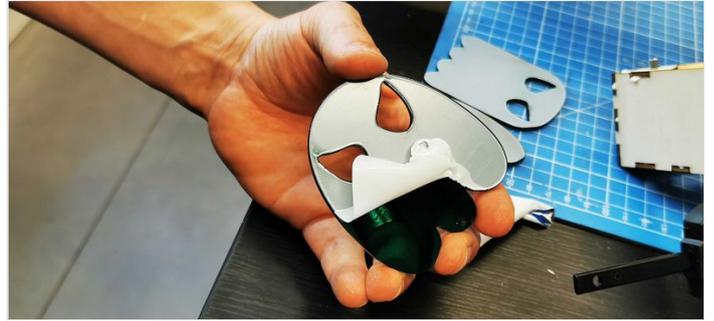
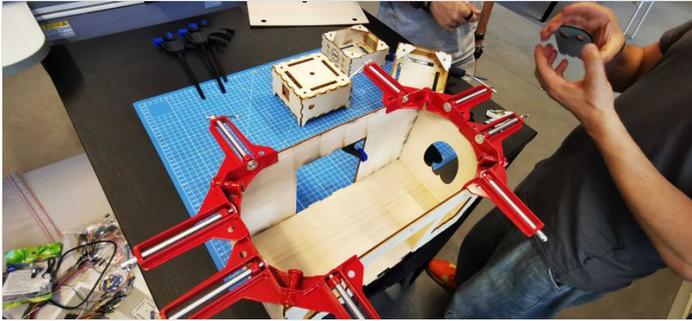
2. Découpage et collage des morceaux

- Bien penser à supprimer les écritures permettant d'identifier les faces avant la découpe
- Bois utilisé : contreplaqué peuplier 3 mm
- Réglages utilisés sur la PerezCamp 140 W
 - puissance max : 40 %
 - puissance min : 30 %
 - vitesse : 40 mm / s
- L'assemblage est expliqué avec les photos BentoGhost02 à BentoGhost04
 - Coller les éléments avec de la colle à bois
- *Le plancher qui supporte le mécanisme est décrit à l'étape suivante*

3. Découpe des fantômes, dans du plexiglas 3 mm

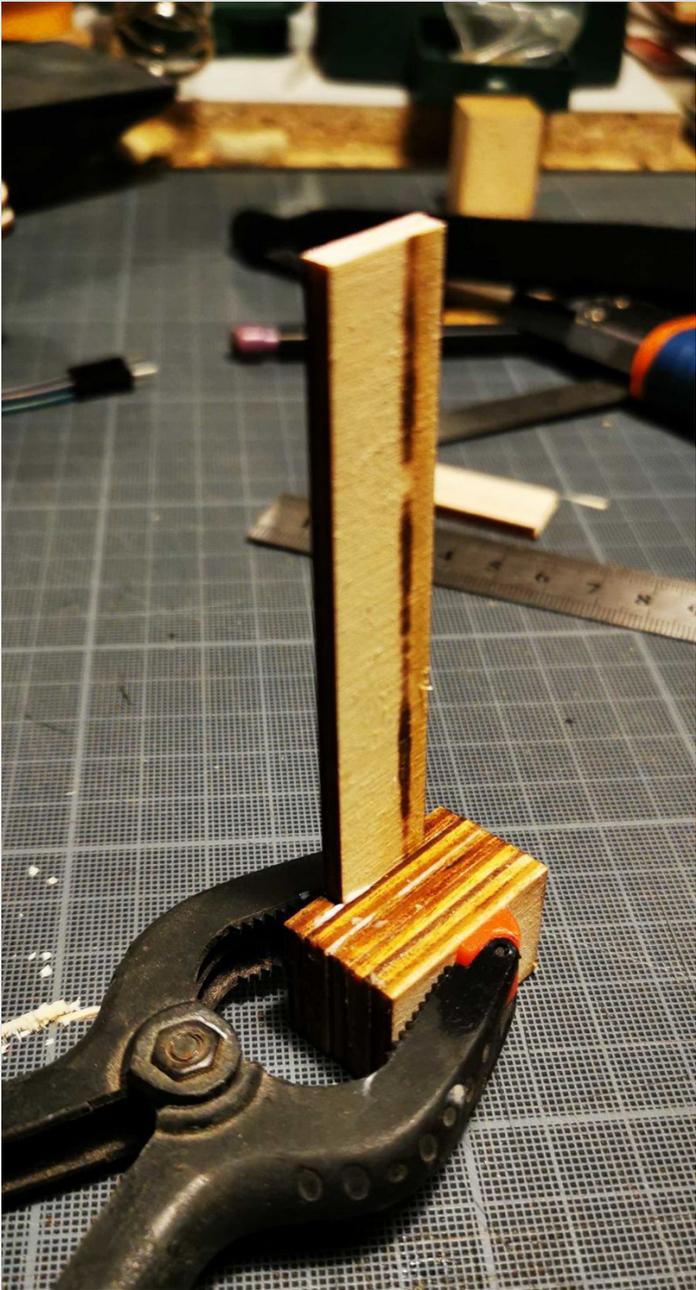
- Plan à télécharger : BentoGhost_fantomes.svg (clic droit / enregistrer le fichier sous)
- Remarque : les 4 fantômes découpés dans la boîte sont en haut de dessin, mais vous pouvez sélectionner ceux que vous voulez à partir du moment où ils sont de la même forme que ceux de la boîte
- L'assemblage est expliqué avec les photos BentoGhost05 à BentoGhost07
 - Remarque : les fantômes peuvent également être collés à la colle à bois. Il faut bien les nettoyer tout de suite pour ne pas avoir de résidus de colle sur le plexiglas.



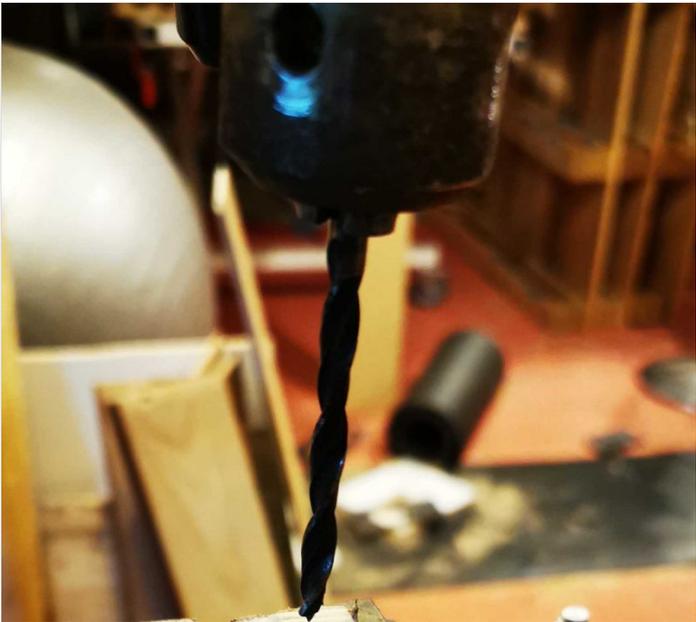


Étape 2 - Fabrication du plancher

- Télécharger le fichier BentoGhost_cibles_et_servo.svg (clic droit / enregistrer le fichiers sous) pour découper toutes les petites pièces qui vont servir à l'assemblage des portes-cible et de toutes les parties du plancher
- BentoGhost09 et BentoGhost10 : assemblage des 3 portes-cible
- BentoGhost12 : perçage d'un trou dans la partie basse des portes-cible
- BentoGhost13 : coller ensuite le fantôme et un lest (nous avons mis un gros écrou)
- Coller les autres petits éléments pour fabriquer les cales qui vont accueillir la tige filetée
 - BentoGhost14 : puis les percer (bien droit)
 - BentoGhost15 : ça ressemblera à ça



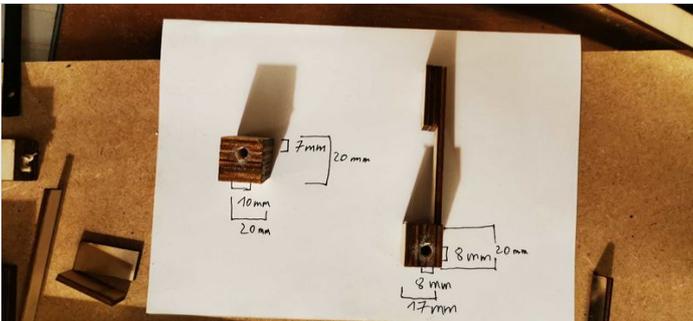






Étape 3 - Fabrication du plancher (suite 1)

- BentoGhost16 : Vous avez alors plusieurs éléments : les 2 cales de côté, et les 3 portes-cible, dont les dimensions sont indiquées sur la photo
- BentoGhost17 et BentoGhost18 : visser la pale qui va entraîner le remonte-cible en étant branchée sur le servomoteur
- BentoGhost19 : coller les 3 éléments du remonte-cible pour obtenir, sur les photos BentoGhost20 et 21, tous les éléments préconfigurés



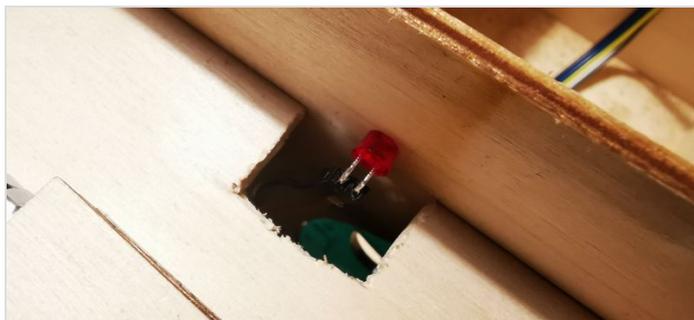






Étape 4 - Fabrication du plancher (suite 2)

Nous avons dû découper le plancher pour laisser passer les leds et ainsi rendre totalement amovible la partie supérieur de la BentoGhost. En outre l'accès à la partie interne, propre au principe de la Bentolux, est respecté.



Étape 5 - Fabrication du plancher (suite 3) porte-cible

- Scier la tige filetée entre deux écrous pour reconstituer le filetage (plus pratique si l'on souhaite visser depuis les deux extrémités de la tige). La tige doit faire 30 cm de long.
- Insérer une cale puis une rondelle et deux écrous, puis deux écrous et une rondelle, ensuite la première cible, une rondelle et deux écrous etc... jusqu'à la cale opposée.
- coller les cales à 14 mm de la partie frontale de la box .
- Ajuster l'espacement des portes-cible et leur centrage par rapport à l'ensemble.
- Coller les écrous de 10 mm pour lester la cible.
- Coller les fantômes.
- Comme vous pouvez le voir sur les photos, l'écrou de 10 et la longueur des portes-cible ont été des réajustements, ceci explique le fait que vous ne les voyez pas sur les photos des fantômes.



Étape 6 - Fabrication du plancher (suite 4) remonte-cible

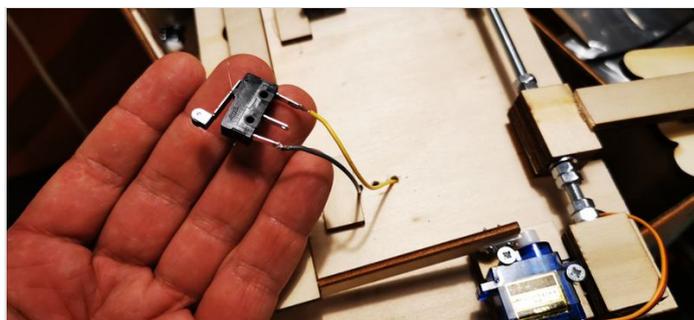
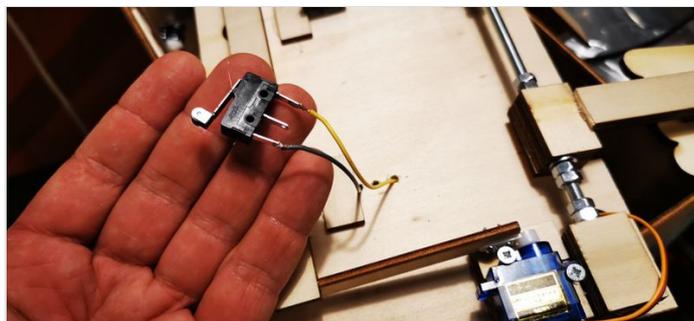
- Avant collage réaliser un trou dans la cale qui va recevoir le clou dans l'axe du servomoteur. Le clou doit être serré dans la cale et libre sur le bras du remonte-cible.
- Réalisez un pré-perçage (pour éviter de fendre la cale) avec un foret de plus petit diamètre que votre clou .
- Coller les cales du remonte-cible contre celles du porte-cible (3 épaisseurs coté servomoteur soit 9mm et 6 épaisseurs de l'autre côté soit 18 mm).
- Faites attention à bien aligner l'axe du servomoteur avec le clou.
- Nous avons provisoirement fixé le servomoteur avec deux vis, pour ce faire nous avons rajouté une cale de 9mm pour une des deux vis.
- **IMPORTANT** avant de fixer le servomoteur, vous devez initialiser et connaître sa position. Un code simple est disponible dans les fichiers à télécharger. (Merci Gael ;-))



Étape 7 - Fabrication du plancher (suite 5) Les 3 microrupteurs

Pour informer Arduino de l'état des cibles nous avons installé des microrupteurs.

- Percer avec un foret de 2mm les trous correspondants aux pattes des microrupteurs.
- Au préalable vous aurez vérifié leur bon alignement avec les cibles.
- Dans les ajustements que nous avons réalisés, nous avons collé des cales de 3mm pour redonner de l'angle aux microrupteurs (vous l'aurez compris, il vaut mieux avoir des cales en rabes... ;-)



Étape 8 - Fabrication du plancher (suite et fin) rampe d'appui des cibles

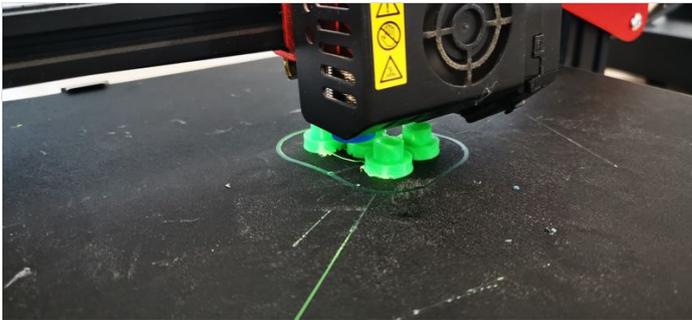
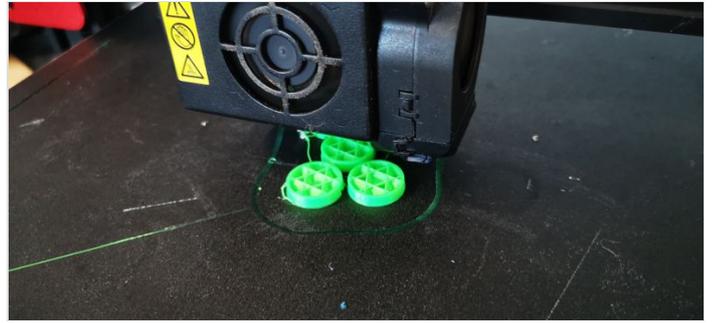
- Pour régler l'inclinaison et la bonne tenue des cibles, vous aurez besoin de réaliser une rampe d'appui, ici elles ont été réalisées en MDF 10 mm de récup. La pièce fait 340x27x10 mm. Nous avons collé des cales de sorte que la rampe puisse être bien fixée.
- Utilisez de préférence des aimants pour vous assurer de la bonne tenue des cibles. Ici les aimants ont été collés avec de la colle forte et maintenue à l'aide de petits serre-joints.



Étape 9 - Fabrication en 3D du bouton potar

Dans cette version de notre projet nous n'utilisons pas de potentiomètre dans notre programme . Toutefois il est branché et nous avons réalisé le bouton qui lui est dédié.

- Prise de cotes au pied à coulisse
- Création dans Fusion 360, puis passage sous Cura pour l'impression. Le fichier est disponible dans les pièces à télécharger.

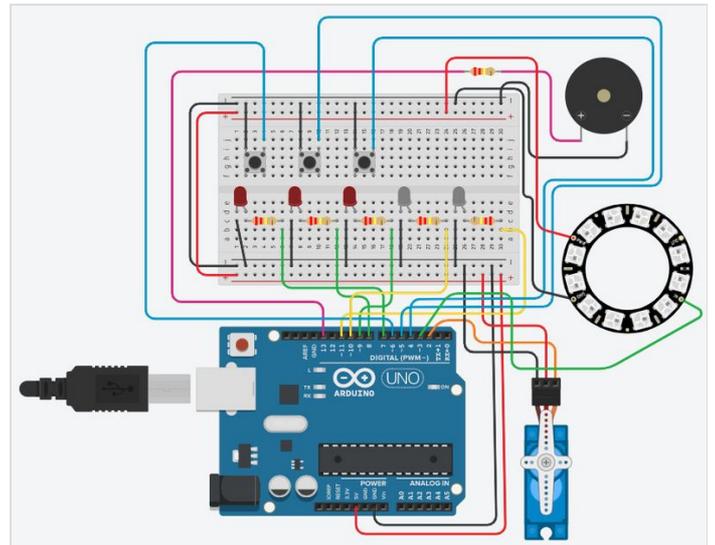
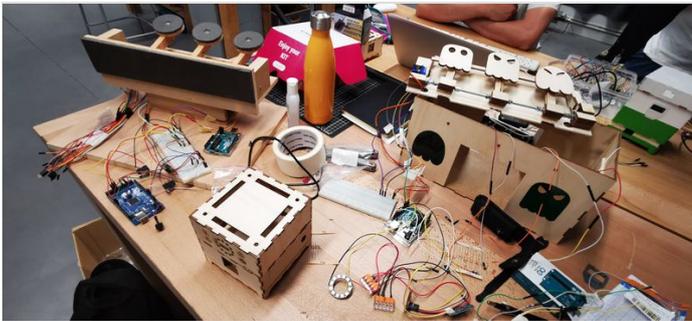


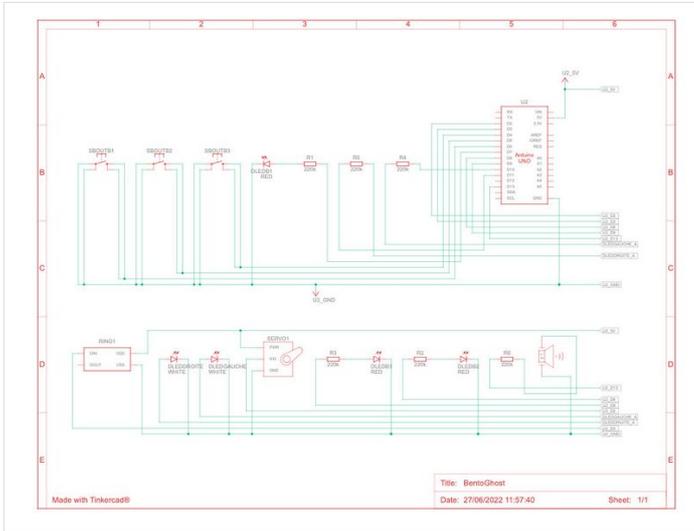
Étape 10 - Branchement électrique Arduino

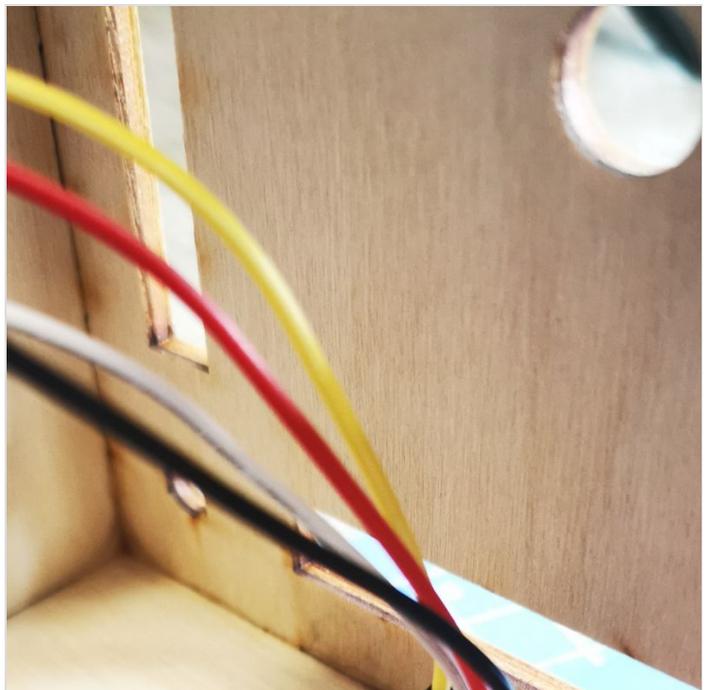
Visualisation Tinkercad des branchements qui correspondent au troisième étage de la Bentolux.

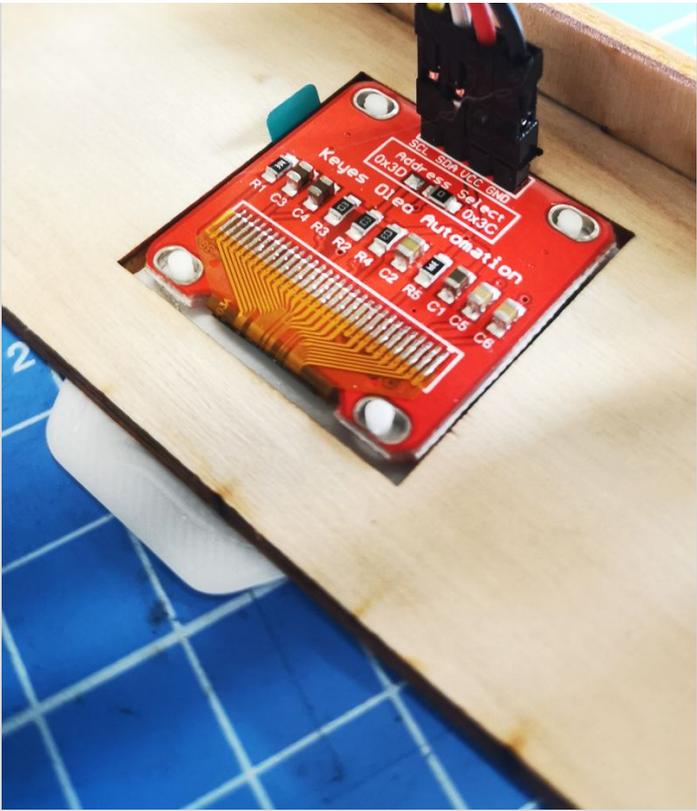
L'écran OLED SSD1306 que nous avons utilisé ne figure pas dans Tinkercad.

Nous l'avons branché ainsi : (A5-SCL)-(A4-SDA)-(VCC-5V)-(GND-GND)

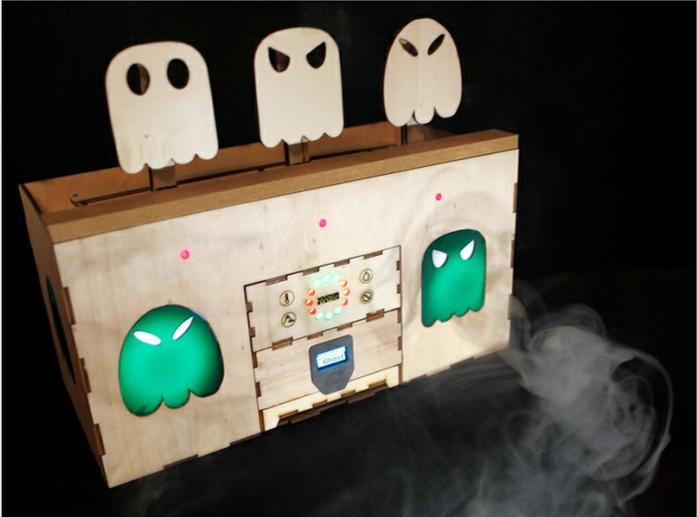








Étape 11 - BOOOooUUUUUUuuuuuu



Étape 12 - Un peu de fun avant le code ;-)



Étape 13 - Programme Arduino

Initialement nous étions partis pour faire trois fois plus gros avec trois fois plus de cibles... avec plein de séquences de jeu...un mode furieux un autre plus méditatif... Bref on s'est vite rendu compte (et sur les bons conseils d'Hubert) qu'il valait mieux commencer avec une structure et un programme simple et pour la suite nous verrons. Arduino nous a bien aidé aussi à calmer nos ardeurs...

Nous avons donc créé un programme simple d'une partie de 60 secondes.

- compte à rebours bip bip bip
- quand les 3 cibles sont tombées elles se relèvent
- dès que les trois cibles sont tombées +3 points
- décompte sonore et visuel les 5 dernières secondes de la partie
- Fin de la partie
- affichage du score sur l'écran pendant 7 secondes -
- Reboot logiciel si on a pas envie d'aller voir le score et qu'on veut continuer à canarder ;-)
- le tout avec des petits effets sonores et visuels (pour la v2 on prévoit une machine à fumée)

```
/* Ci-dessous, retrouvez le programme de Renaud et Mayak permettant d'animer
la BentoGhost réalisé dans le cadre de la formation FabNumAura en 2022.
Si vous décidez de marcher dans nos pas, pour vous, vos enfants ou vos parents, n'hésitez pas à
nous envoyer des photos de vos versions.
Bonne lecture et bonne continuation dans vos projets.
*/
```

```
#include <avr/wdt.h> // bibliothèque pour reboot le game
```

```
//////////////////// Ring Led //////////////////////
```

```
#include <FastLED.h>
```

```
#define NUM_LEDS 12
```

```
#define DATA_PIN 3
```

```
#define CLOCK_PIN 13
```

```
CRGB leds[NUM_LEDS];
```

```
//////////////////// SERVOS //////////////////////
```

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

```
Servo myservo1;
```

```
int score = 0; // point par cible
```

```
unsigned long tpsPartie = 65000; // le tps que la bête se réveille il y a déjà 5 secondes donc 65000 ms pour une partie d'une minute
```

```
unsigned long tpsEnCours = 0; // variable qui sera associé à millis pour gérer le temps de la partie
```

```
int pos90 = 102; // variable indiquant la position du servomoteur à 90° (à ajusté ici 102°)
```

```
int pos0 = -2; // variable indiquant la position du servomoteur à 0° (à ajusté ici -2°)
```

```
////////////////////POTAR START GAME////////////////////
```

```
//const int potarStartGame = A2; // le potar qui active le jeu est sur la broche analogique A2, nous l'utiliserons pour une autre version
```

```
//////////////////// BUZZER////////////////////
```

```
const int pinBuzzer = 13; //le buzzer est sur la broche 13
```

```
//////////////////// STAND de TIR////////////////////
```



```

// on éclaire les 3 led rouges en façade
digitalWrite(ledB1, HIGH);
digitalWrite(ledB2, HIGH);
digitalWrite(ledB3, HIGH);
delay(1000);
// on éclaire les 2 led vertes en façade
digitalWrite(ledGauche, HIGH);
digitalWrite(ledDroite, HIGH);

myservo1.attach(2); // le servomoteur myservo1 est relié à la broche 2
// le servo remonte les cibles puis redescend
myservo1.write(pos90);
Serial.println("pos90");
delay(1000);
myservo1.write(pos0);
Serial.println("pos0");
delay(1000);

}
void loop()
{

gameBentoGhost(); //Fonction de la partie de jeu
scoreFinal(); //Fonction permettant l'affichage du score de la partie

}
//La partie de jeu
void gameBentoGhost()
{

while (tpsEnCours < tpsPartie) // tant que le temps en cours est inférieur au temps total de la partie
{
tpsEnCours = millis(); // on associe la variable tpsEnCours avec la fonction millis qui compte le temps dès le lancement de l'arduino
cibleBoutServo(); // on appelle la fonction cibleBoutServo qui gère le comportement des cibles, des microrupteurs et du servomoteur
Serial.println(score); //pour contrôler que tout se passe bien dans le moniteur série au niveau du score
Serial.println(tpsEnCours); //pour contrôler que tout se passe bien dans le moniteur série au niveau du temps

if (millis() > 59000) // à 5 seconde de la fin de la partie
{
myservo1.detach(); // on détache le servomoteur pour qu'il ne relève plus les cibles et accessoirement ils se mettent à trembler un peu à ce moment là
// petits effets sons et lumière de fin de partie :-
tone(pinBuzzer, 300, 500);
digitalWrite(ledB1, LOW);
digitalWrite(ledB2, LOW);
digitalWrite(ledB3, LOW);
delay(200);

tone(pinBuzzer, 600, 500);
digitalWrite(ledB1, HIGH);
digitalWrite(ledB2, HIGH);
digitalWrite(ledB3, HIGH);
delay(200);

tone(pinBuzzer, 300, 500);
digitalWrite(ledB1, LOW);
digitalWrite(ledB2, LOW);
digitalWrite(ledB3, LOW);

delay(200);
tone(pinBuzzer, 600, 500);
delay(200);
tone(pinBuzzer, 300, 500);

if (tpsEnCours >= tpsPartie) // si le temps en cours est égal ou supérieur à la partie
{
finalRing(); // petits effets sons et lumière
finalRing();
finalRing();
digitalWrite(ledGauche, LOW);
digitalWrite(ledDroite, LOW);
finalRing();
finalRing();
finalRing();
digitalWrite(ledGauche, HIGH);

```



```

    leds[11] = CRGB::Black;
    FastLED.show();
}
}
void finalRing() // même fonction que la précédente sauf que l'anneau s'éclaire en GREEN (donc rouge) pour la fin de partie
{
    FastLED.setMaxPowerInVoltsAndMilliamps(5, 10); // réglage de la puissance des led du Ring
    FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);

    for (int redLed = 0; redLed < NUM_LEDS; redLed = redLed + 1) {
        // Turn our current led on to white, then show the leds
        leds[redLed] = CRGB::Green;

        // Show the leds (only one of which is set to white, from above)
        FastLED.show();

        // Wait a little bit
        delay(50);

        leds[redLed] = CRGB::Black;
        leds[11] = CRGB::Black;
        FastLED.show();

    }
}

void scoreRing() // meme fonction que la précédente en BLEUE qui pour le coup est bleu, utilisé avant affichage du score final
{
    // Move a single blue led
    FastLED.setMaxPowerInVoltsAndMilliamps(5, 10); // réglage de la puissance des led du Ring
    FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);

    for (int blueLed = 0; blueLed < NUM_LEDS; blueLed = blueLed + 1) {
        // Turn our current led on to white, then show the leds
        leds[blueLed] = CRGB::Blue;
        leds[11] = CRGB::Black;
        // Show the leds (only one of which is set to white, from above)
        FastLED.show();

        // Wait a little bit
        delay(300);

        // Turn our current led back to black for the next loop around
        leds[blueLed] = CRGB::Black;
    }
}

// tiré d'un exemple de ScrollText d'adafruit, il a fallu changer
// tous les display par écranOLED pour que ça fonctionne et jouer avec les positions de curseur pour le texte
// un lien qui m'a beaucoup aidé pour l'écran https://passionelectronique.fr/ecran-oled-i2c-arduino/
void scoreFinal()
{
    scoreRing();
    écranOLED.clearDisplay();

    écranOLED.setTextSize(3); // Draw 2X-scale text
    écranOLED.setTextColor(SSD1306_WHITE);
    écranOLED.setCursor(38, 5); // ici on joue avec les axes x et y pour positionner le texte de la ligne suivante
    écranOLED.println(F("SCORE")); // le texte que l'on souhaite voir apparaître à l'écran
    écranOLED.setCursor(73, 33); // ici on joue avec les axes x et y pour positionner le texte de la ligne suivante
    écranOLED.println(score); // l'affichage de la valeur de notre résultat de partie
    //écranOLED.println(score);
    écranOLED.display(); // Show initial text
    delay(100);

    // Scroll in various directions, pausing in-between:
    écranOLED.startscrollright(0x00, 0x0F);
    delay(2000);
    écranOLED.stopscroll();
    delay(1000);
    écranOLED.startscrollleft(0x00, 0x0F);
    delay(2000);
    écranOLED.stopscroll();
    delay(1000);
    écranOLED.startscrollright(0x00, 0x0F);

```

```

ecranOLED.startscrollright(0x00, 0x07),
delay(2000);
ecranOLED.startscrollleft(0x00, 0x07);
delay(2000);
ecranOLED.stopscroll();
delay(7000); // Ici on peut changer le temps d'affichage du résultat de la partie

software_Reboot(); // et là on REBOOT
}

void imageBentoGhost()
/// fonction permettant l'affichage d'une image le lien pour générer le code https://javl.github.io/image2cpp/?pseSrc=pgEcranOledArduino
{
  if (!ecranOLED.begin(SSD1306_SWITCHCAPVCC, adresseI2CecranOLED))
    while (1); // Arrêt du programme (boucle infinie) en cas d'échec de l'initialisation

  // Affichage d'une image au centre de l'écran
  ecranOLED.clearDisplay(); // Effaçage de la mémoire tampon de l'écran OLED

  ecranOLED.drawBitmap(
    (ecranOLED.width() - largeurDeLimage) / 2, // Position de l'extrême "gauche" de l'image (pour centrage écran, ici)
    (ecranOLED.height() - hauteurDeLimage) / 2, // Position de l'extrême "haute" de l'image (pour centrage écran, ici)
    epd_bitmap_BentoGhost,
    largeurDeLimage,
    hauteurDeLimage,
    WHITE); // "couleur" de l'image

  ecranOLED.display(); // Transfert de la mémoire tampon à l'écran OLED, pour affichage
}

void software_Reboot() // fonction permettant de redémarrer l'arduino en passant par le logiciel
{
  wdt_enable(WDTO_15MS);

  while (1)
  {

  }
}

```