



Makers game

Table d'arcade pour jouer à Pong, Snake ou Tetris entre amis.

 Difficulté **Difficile**

 Durée **2 jour(s)**

 Catégories **Art, Mobilier, Jeux & Loisirs**

 Coût **300 EUR (€)**

Sommaire

Introduction

Video d'introduction

Étape 1 - Découper les pièces

Étape 2 - Poncer les pièces

Étape 3 - Assembler la table

Étape 4 - Mettre en place le bandeau

Étape 5 - Préparer et installer le ruban de LED

Étape 6 - Programmer la Raspberry Pi 1

Étape 7 - Installer les boutons d'arcade

Étape 8 - Réaliser le montage électronique

Étape 9 - Branchements des 10 boutons sur la Raspberry Pi

Étape 10 - Programmation du jeu "Pong" - JavaScript

Étape 11 - Programmation du jeu "Snake" - Python

Étape 12 - Mettre un léger point de colle avec le pistolet à colle sur les fils extérieure à la grille

Étape 13 - Réalisation du contour de la table

Étape 14 - Coller le flexwood

Étape 15 - Percer un trou pour faire passer les fils électriques

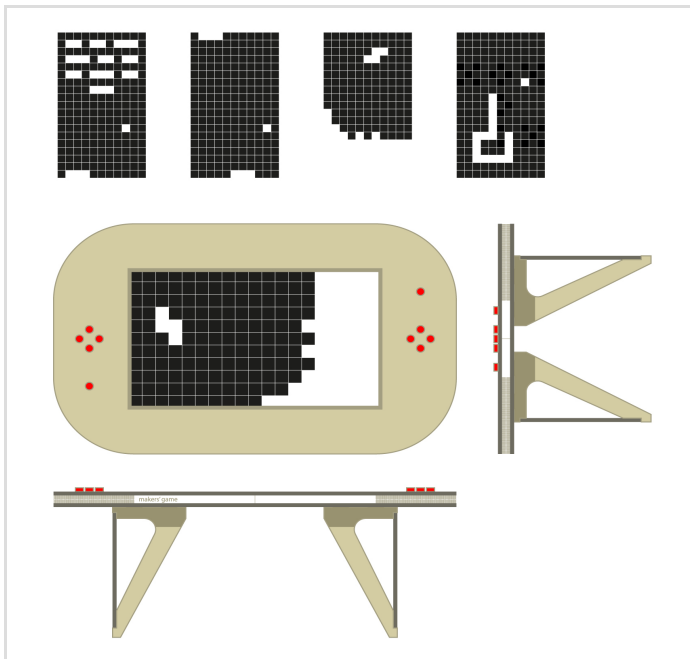
Étape 16 - Connexion des 10 boutons avec les GPIO

Étape 17 - Installation des éléments électroniques de la table

Commentaires

Introduction

Le rétro-gaming, c'est le pied ! Débarrassez la table basse de l'apéro des habituels jeux de société, et retrouvez Pong, Snake et Tetris dans une dimension inattendue.



Matériaux

- contre-plaqué d'épaisseur 10mm
- médium d'épaisseur 3mm
- Arduino, carte epoxy
- Raspberry Pi 1
- ruban de LED

Outils

- découpeuse laser
- pistolet à colle, papier à poncer, perceuse-visseuse, fer à souder

🔗 Rocket kit

📄 Table arcade bandeau.svg

📄 Table arcade dessous.svg

📄 Table arcade dessus.svg

📄 Table arcade pieds.svg

📄 Table arcade quadrillage interieur.svg

📄 Table arcade vitre.svg

Étape 1 - Découper les pièces

Dans l'onglet fichier, **télécharger les documents**. Et découper à la laser :

En **contre-plaqué de 10mm d'épaisseur** :

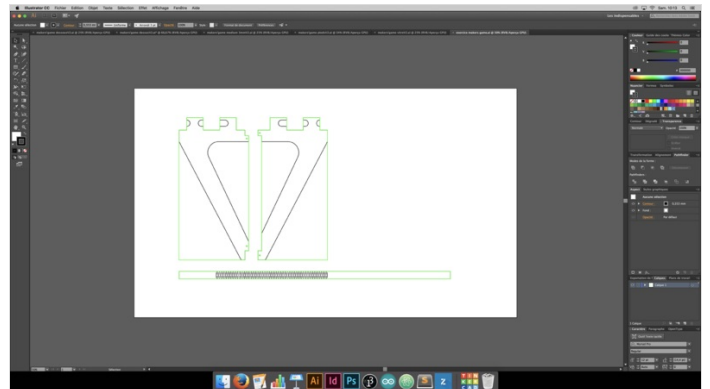
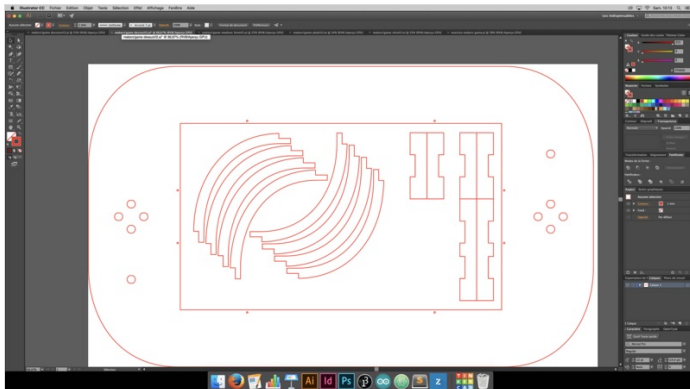
- pieds de la table (x4)
- plateau du dessous (x1)
- plateau du dessus avec pièces annexes (x1) Poncer les rebords des découpes afin d'obtenir un rendu propre.

En **médium de 3mm d'épaisseur**:

- 4 x bandeaux contours de la table (médium)
- 1 x quadrillage intérieur (x1)

En **PMMA**:

- vitre de la table (x1)





Étape 2 - Poncer les pièces

Nettoyer et poncer les pieds pour enlever les traces de brûlures. Pour poncer, faites des mouvements circulaires dans le sens du bois.






Étape 3 - Assembler la table

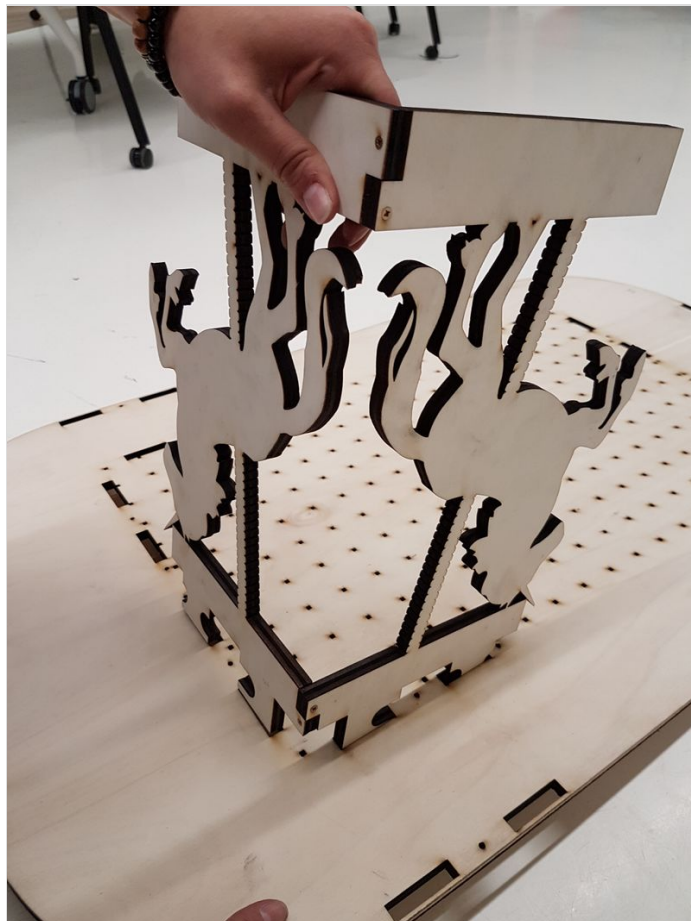
Assembler les pieds :

- Nettoyer soigneusement les trous taraudés des vis.
- Visser les planches de bois deux à deux (vis de 2,5 x 20mm)
- Insérer les 4 pieds dans le plateau du dessous et visser

 Attention à ne pas abimer les encoches !

Assembler la grille :

- Vous munir de patience !
- Les lattes verticales et horizontales doivent être bien alignées une fois enfoncées.



Étape 4 - Mettre en place le bandeau

Emboîter les 6 cales de CP 10mm aux emplacements sur le plateau du dessous et les fixer avec des vis à bois.

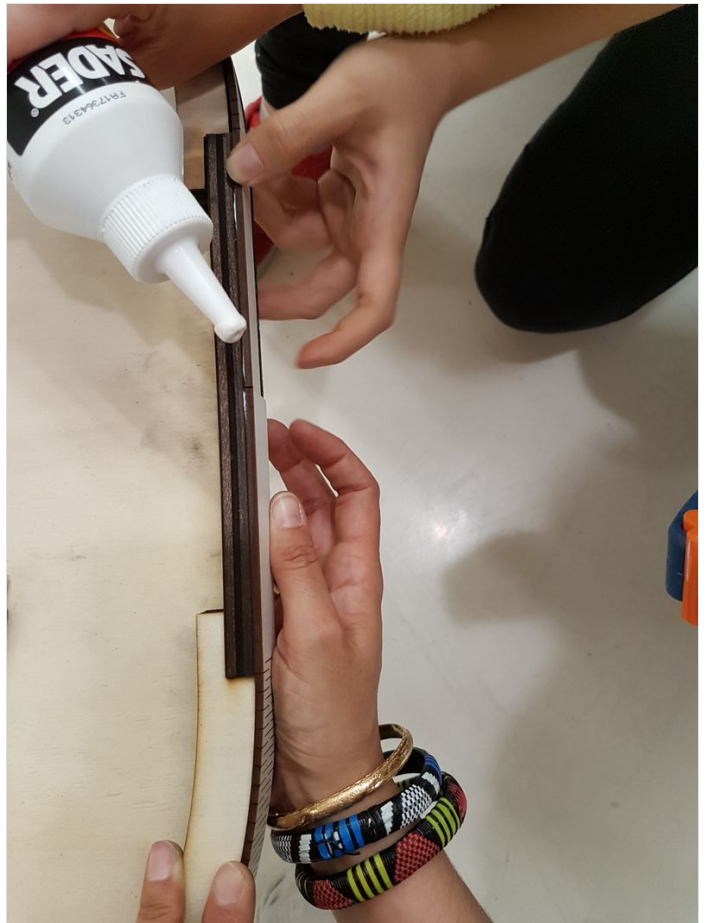
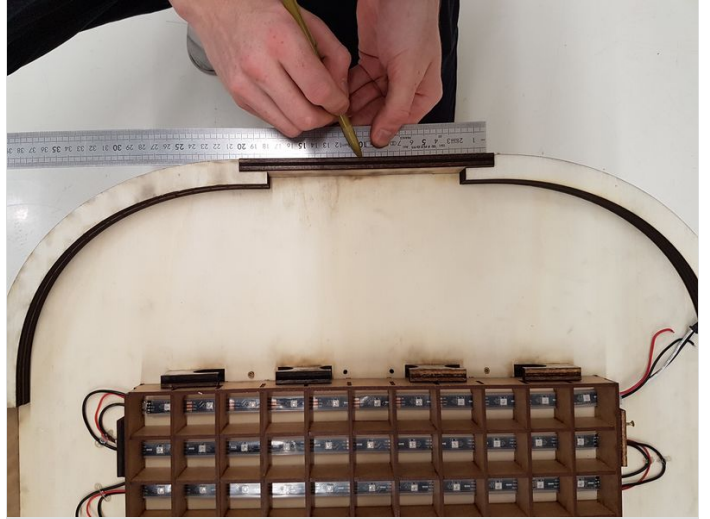
Une fois les cales vissées, placer les 4 pièces découpées en amont pour les angles arrondis de la table.

Trouver le milieu de la largeur de chaque bandeau et le marquer sur la cale en bois.

Placer le flexwood à partir du milieu de la cale en bois.

Coller le flexwood (mettre de la colle à bois sur les parties où il y a les cales).

Pour rejoindre les 2 bouts de flexwood : collez un petit bout de medium 3mm permettant de les réunir et de solidifier l'ensemble.





Étape 5 - Préparer et installer le ruban de LED

Nous avons utilisé le ruban de leds Adafruit Neopixel (11 x 19 = 209 leds).

Couper votre ruban en **19 rubans de 11 leds**.


Souder les extrémités de vos rubans de façon à réaliser un « serpent » (cf. photo): **GND - GND / DIN - DOUT / 5V - 5V**. Attention: la longueur de vos fils doit être suffisante pour pouvoir disposer les leds comme sur la photo.

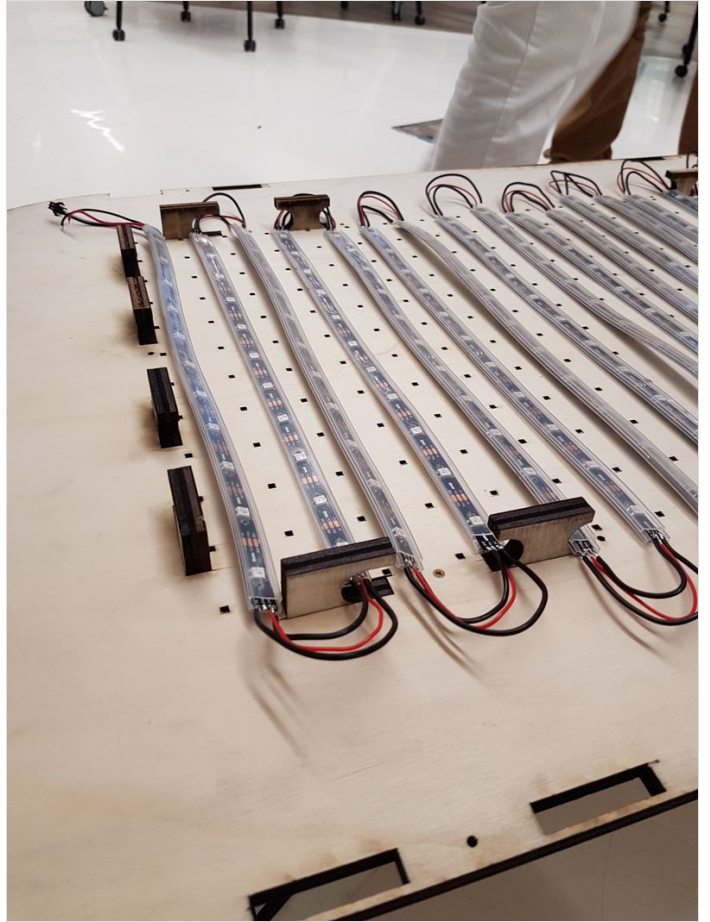
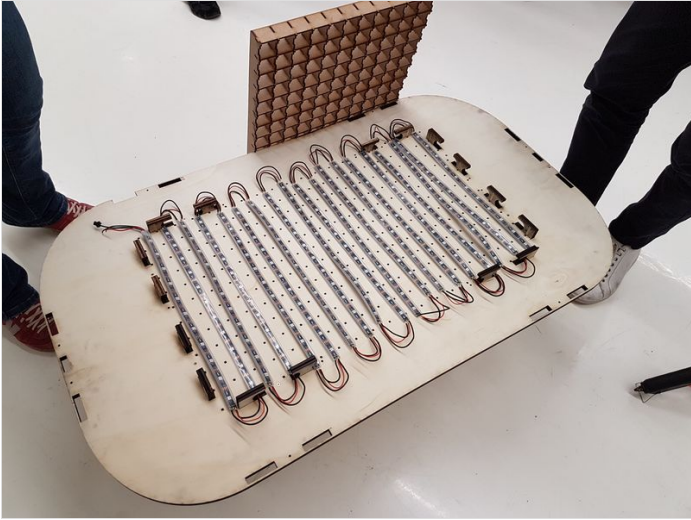
Une fois les soudures réalisées, les fixer avec un point de pistolet à colle.

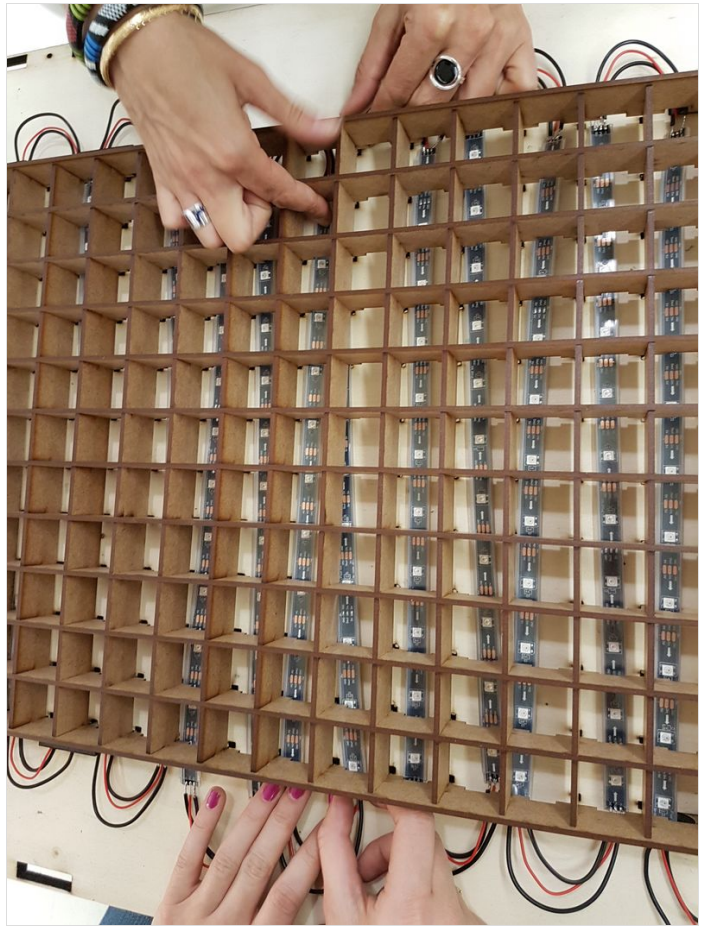
Placer le ruban LED sur la table. Faire en sorte que chaque LED soit à peu près au milieu de chaque case de la grille. Les câbles doivent pouvoir passer entre les encoches supérieures des pieds de la table.

Tester les soudures de votre ruban à l'aide de l'Arduino Uno. Pour cela, nous utilisons l'exemple « Simple » de la librairie « Adafruit Neopixel ».

Une fois testé, disposer parfaitement votre ruban de leds et **enfoncer la grille dans le plateau** du dessous de la table.

 Programme disponible sur Github: [arcadeTable_arduino](#)





Étape 6 - Programmer la Raspberry Pi 1

Installer le système de la Raspberry Pi:

Télécharger et copier l'image de Raspbian Stretch Lite sur votre Raspberry Pi en utilisant Etcher.io.

Vous pouvez suivre le guide installing operating system images pour plus de détails.

Démarrer votre Raspberry Pi avec un clavier, un câble ethernet relié à votre box internet et un écran branchés.

(identifiant par défaut: pi / mot de passe: raspberry)

```
sudo raspi-config
```

 Une fois identifié, vous pouvez taper: pour configurer la langue du clavier (en français:

<http://www.tropfacile.net/doku.php/raspberry-pi/comment-passer-votre-raspberry-en-francais>)

Installez openFrameworks en suivant le guide suivant: Getting your Raspberry Pi ready for openFrameworks

Une fois openFrameworks installé et testé, vous pouvez maintenant télécharger le code des jeux et le compiler :

```
cd /home/pi/openFrameworks/apps/myApps/ git clone
```

 Ne lancez pas le programme

```
https://github.com/emlyon/makers-game-code.git make
```

 maintenant: si l'Arduino n'est

pas branchée, le programme ne peut pas fonctionner.

Pour que le jeu se lance automatiquement au démarrage, éditez le

```
rc.local
```

 fichier :

```
sudo nano /etc/rc.local
```

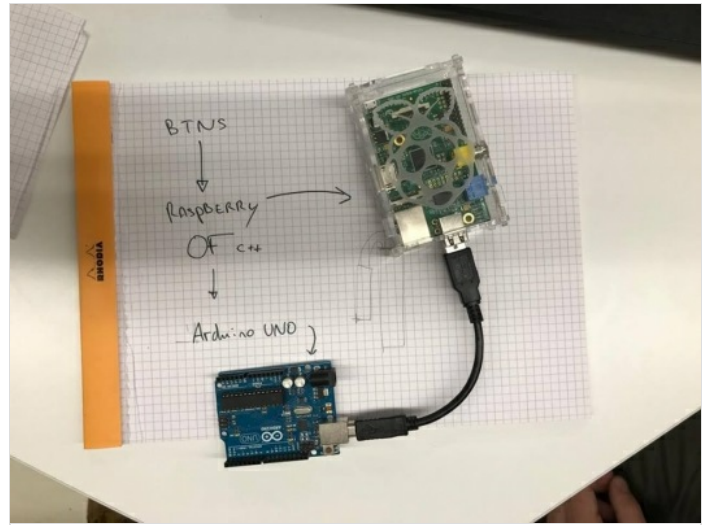
 Et ajouter avant la ligne :

```
exit
```

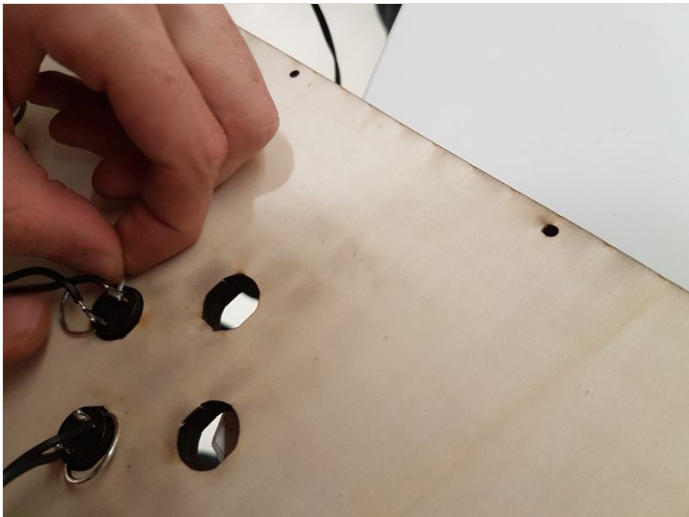
 Faites pour quitter, puis

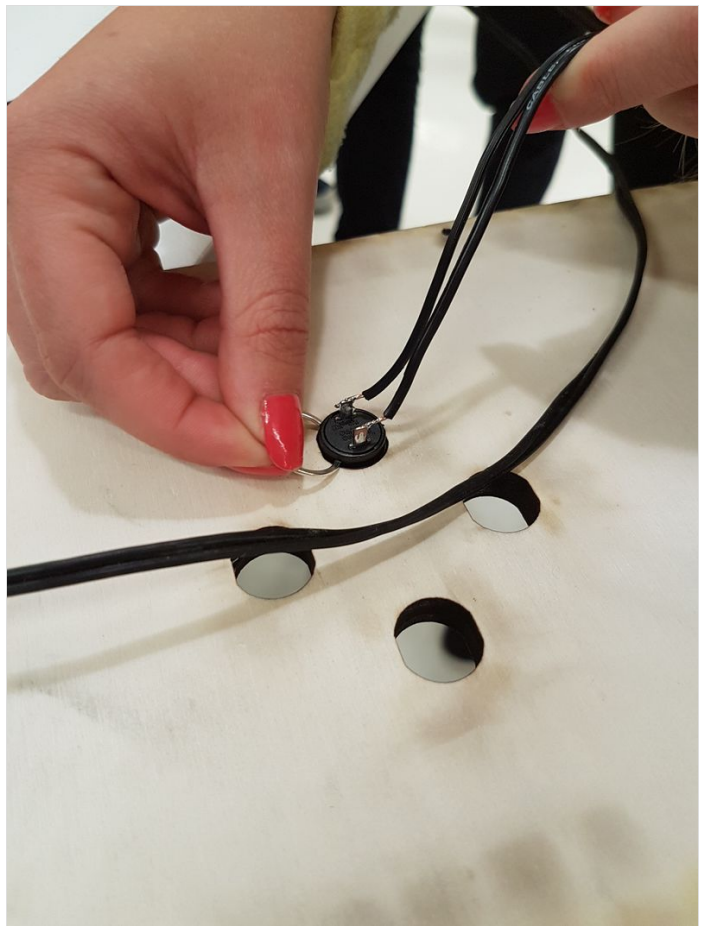
```
su pi -c 'cd /home/pi/openFrameworks/apps/myApps/makers-game-code && make run'
```

```
Ctrl+x
```

 y pour sauvegarder.

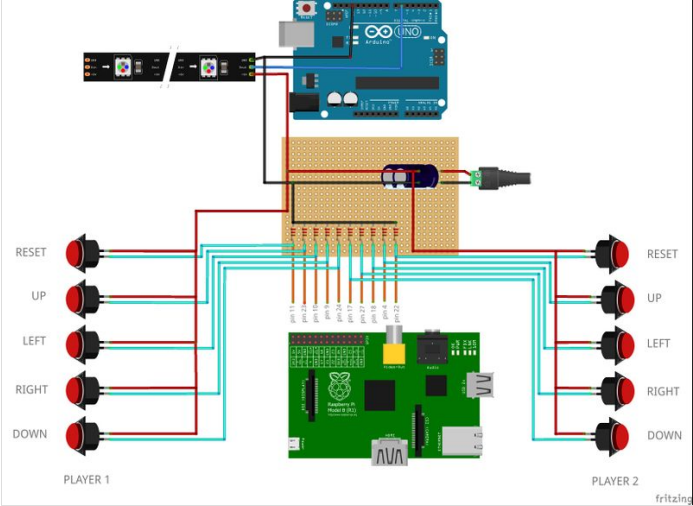
Étape 7 - Installer les boutons d'arcade



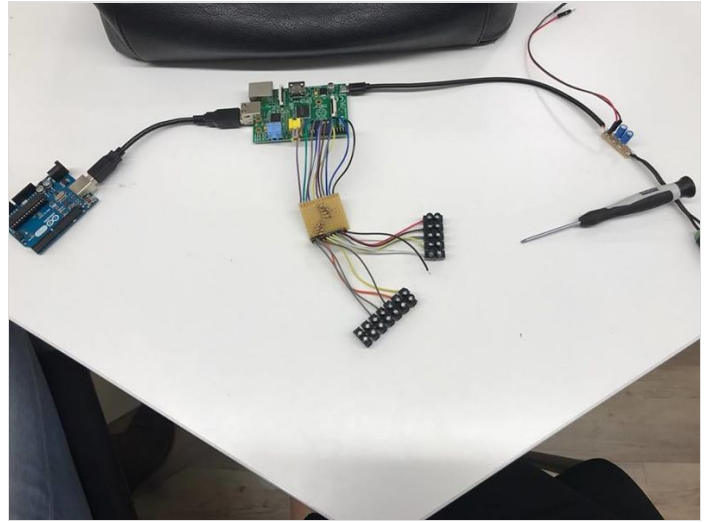
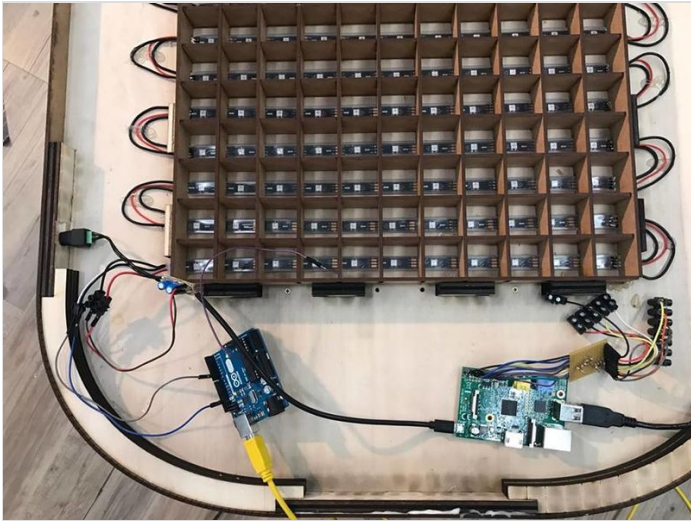




Étape 8 - Réaliser le montage électronique



Étape 9 - Branchements des 10 boutons sur la Raspberry Pi



Étape 10 - Programmation du jeu "Pong" – JavaScript

```
man2p.js
// Définition des différentes variables utilisées pour la programmation du jeu
var started = false;
var hasScored = false;
var win = false;
var score = 0;
var score2 = 0;
var ballSize = 10;
var ballX = 100;
var ballY = 50;
var ballVx = 5;
var ballVy = 5;

// Fonction de la fonction setup (caractéristiques de l'espace de jeu)
function setup() {
  // Dimension et couleur de canvas (101 x 101 de la fenêtre)
  createCanvas(windowWidth, windowHeight);

  // Fonction initiale de la balle
  ballX = width/2;
  ballY = height/2;

  // Fonction initiale des "raquettes"
  pix = width / 2 - largeur / 2;
  pix2 = width / 2 + largeur / 2;

  // Pour jouer contre
  noStroke();
}

// Fonction de jeu
background(255,255,255);
background(255,255,255);
}
```

```
snakey
// Importation des différentes bibliothèques utilisées
import pygame
from pygame.locals import *
from random import randint
import sys

# Size
# squareize = 20
grid = {'width': 45, 'height': 30}

# Coordonnées des pixels
def pixel(x,y):
  pix = ['x', 'y']
  return pix

# Description de la collision
def collision(pix1,pix2):
  if pix1['x'] == pix2['x'] and pix1['y'] == pix2['y']:
    return True
  else:
    return False

# Quitter le jeu et afficher le score
def end():
  font2 = pygame.font.SysFont('Arial', 30)
  endtxt = font2.render("Perdu ! Retente ta chance ! Ton score est de "+str(score), True, (255,255,255))
  screen.blit(endtxt, (10,40))
  pygame.display.update()
  pygame.time.wait(1500)
  sys.exit(0)
}
```

```
man2p.js
// Importer un document HTML
// Permet de régler les marges : col #
//
body{
  margin: 0;
}
}
```

```
js.js
// Déclaration et export de la librairie js.js (en ligne)
// Revoir modification sur le document HTML
//
// js.js v0.1.0 - 10 March 2017
function (global, factory) {
  if (typeof exports === 'object' && typeof module === 'undefined') {
    module = { exports: {} };
    factory(global, module);
  } else {
    factory(global, factory);
  }
}

// Description des process : a tiny implementation of Promises/A+
// description copyright (c) 2014 James Hall, Tom Dale, Steven Pinner and contributors. (Conversion to ES6 API by Jake Archibald)
// License: Licensed under MIT license
// See https://www.npmjs.com/package/promise for details
var Promise = (function (global) {
  function objectOrFunction(x) {
    return typeof x === 'function' || typeof x === 'object' && x !== null;
  }

  function isFunction(x) {
    return typeof x === 'function';
  }

  var _library = undefined;
  if (typeof library !== 'undefined') {
    // ...
  }
}
```

Étape 11 - Programmation du jeu "Snake" – Python

```
snakey
// Importation des différentes bibliothèques utilisées
import pygame
from pygame.locals import *
from random import randint
import sys

# Size
# squareize = 20
grid = {'width': 45, 'height': 30}

# Coordonnées des pixels
def pixel(x,y):
  pix = ['x', 'y']
  return pix

# Description de la collision
def collision(pix1,pix2):
  if pix1['x'] == pix2['x'] and pix1['y'] == pix2['y']:
    return True
  else:
    return False

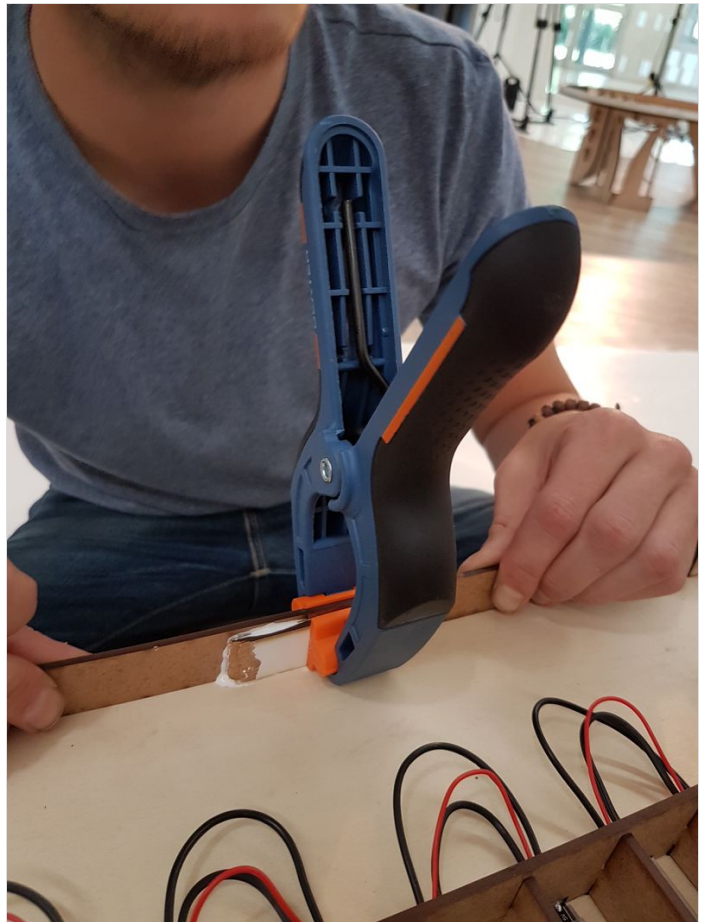
# Quitter le jeu et afficher le score
def end():
  font2 = pygame.font.SysFont('Arial', 30)
  endtxt = font2.render("Perdu ! Retente ta chance ! Ton score est de "+str(score), True, (255,255,255))
  screen.blit(endtxt, (10,40))
  pygame.display.update()
  pygame.time.wait(1500)
  sys.exit(0)
}
```

Étape 12 - Mettre un léger point de colle avec le pistolet à colle sur les fils extérieurs à la grille

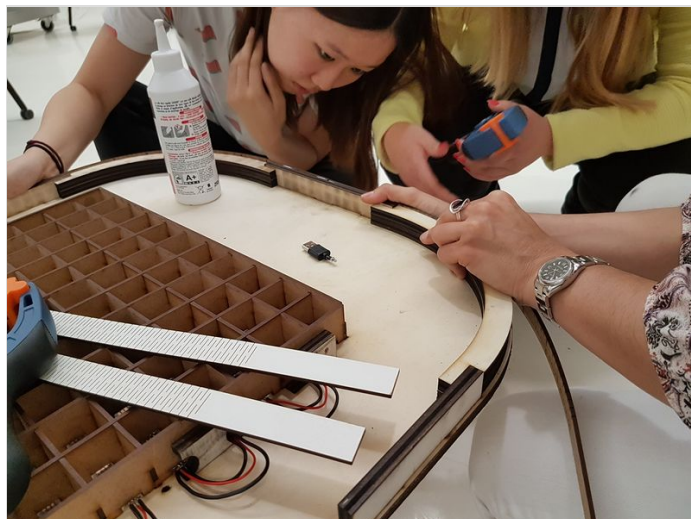


Étape 13 - Réalisation du contour de la table

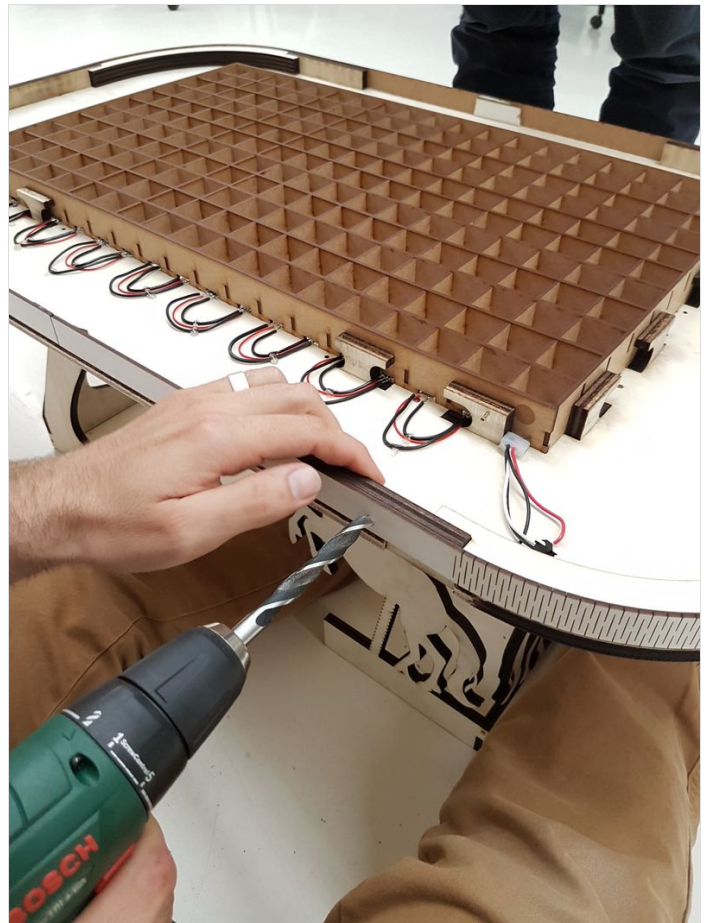




Étape 14 - Coller le flexwood

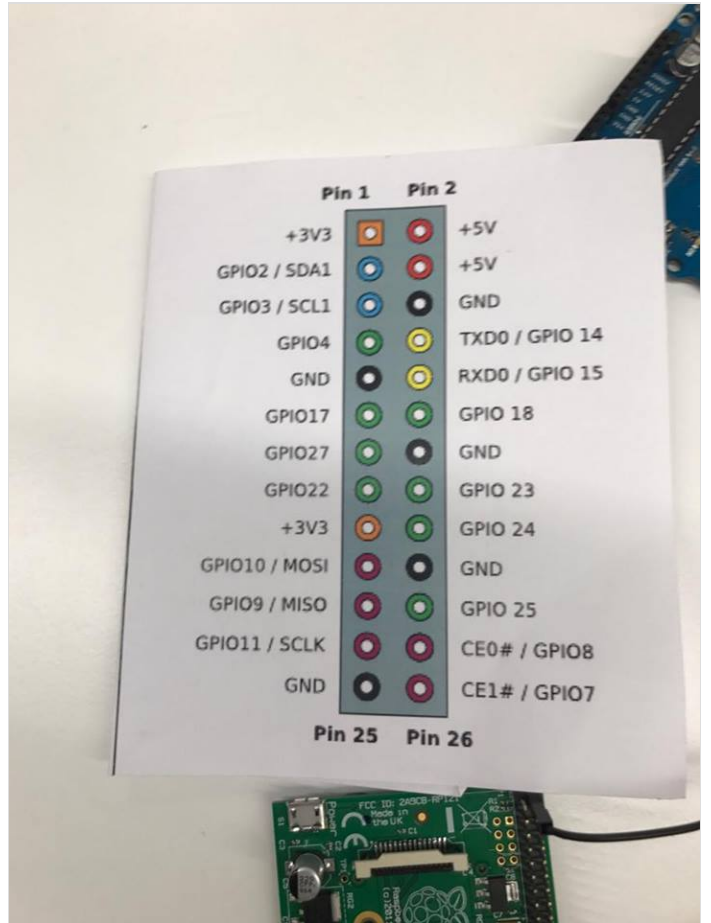
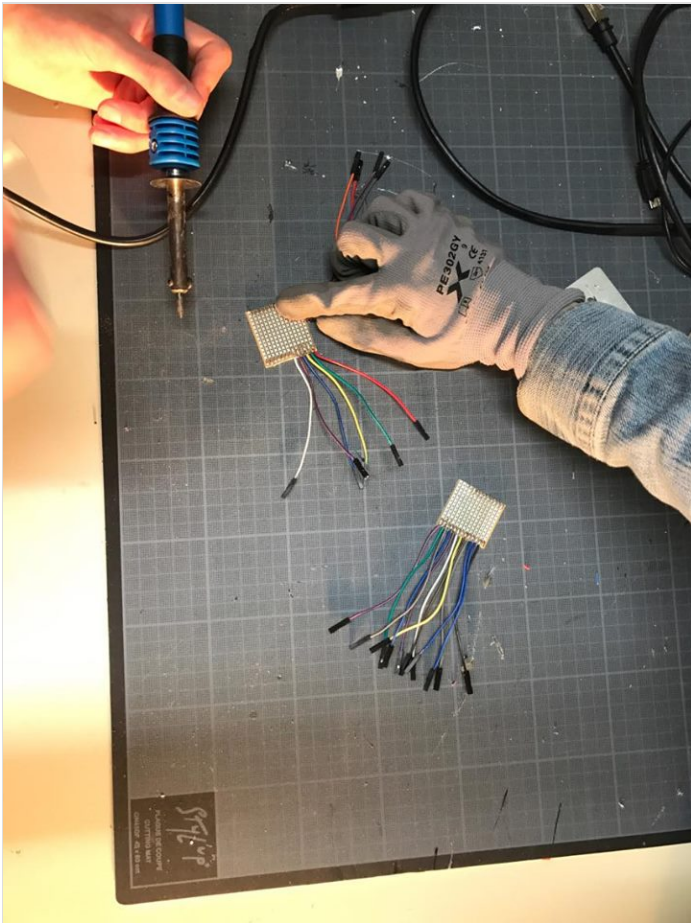


Étape 15 - Percer un trou pour faire passer les fils électriques





Étape 16 - Connexion des 10 boutons avec les GPIO





Étape 17 - Installation des éléments électroniques de la table

