09/10/2019

LASER GAME

Par Skytech

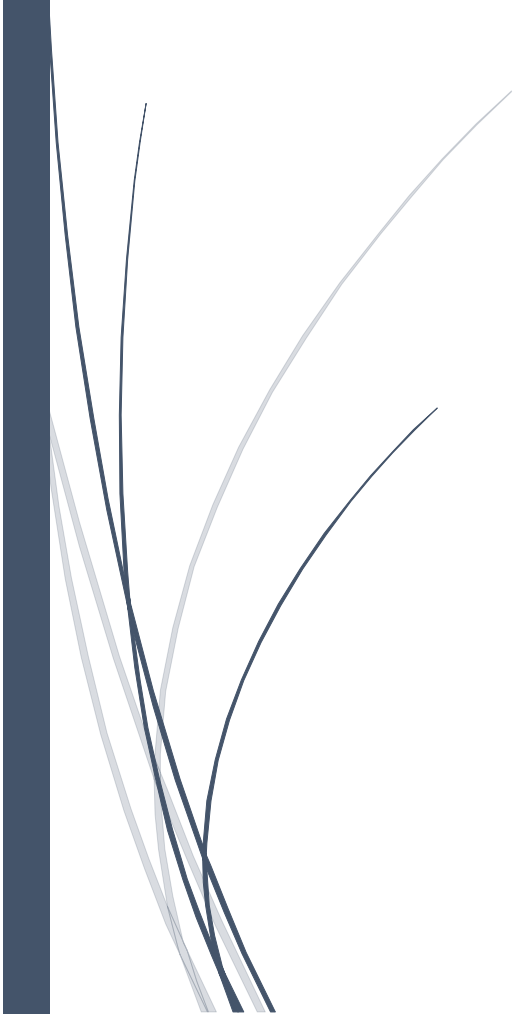


Table des matières

Principe de jeu

Comme un stand de tir, le principe du jeu est de pouvoir toucher les différentes cibles qui s'allumeront de manière aléatoire afin de gagner plusieurs points ! Quoi de mieux pour s'amuser ?

Les cibles

Élite

Les cibles élites de couleur rouge, vous rapportent **20 points**.

Jaune

Les cibles de couleur jaune vous rapportent **10 points**.

Bleu

Les cibles de couleur bleue vous rapportent **5 points**.

Verte

Les cibles de couleur verte vous rapportent **1 point**.

Attention ! car les cibles s'éteignent après un certain temps, soyez donc rapide et précis.

Les différents modes

Mode solo

Marqué le plus haut score afin d'être premier dans le tableau des scores ! Vous serez un tireur d'élite solitaire capable de tous les battre ! **Comment cela se déroule ?** Les différentes NéoPixels s'allume de manière aléatoire ainsi que d'une couleur aléatoire. Le but étant de faire le plus de points possibles.

Mode multijoueur

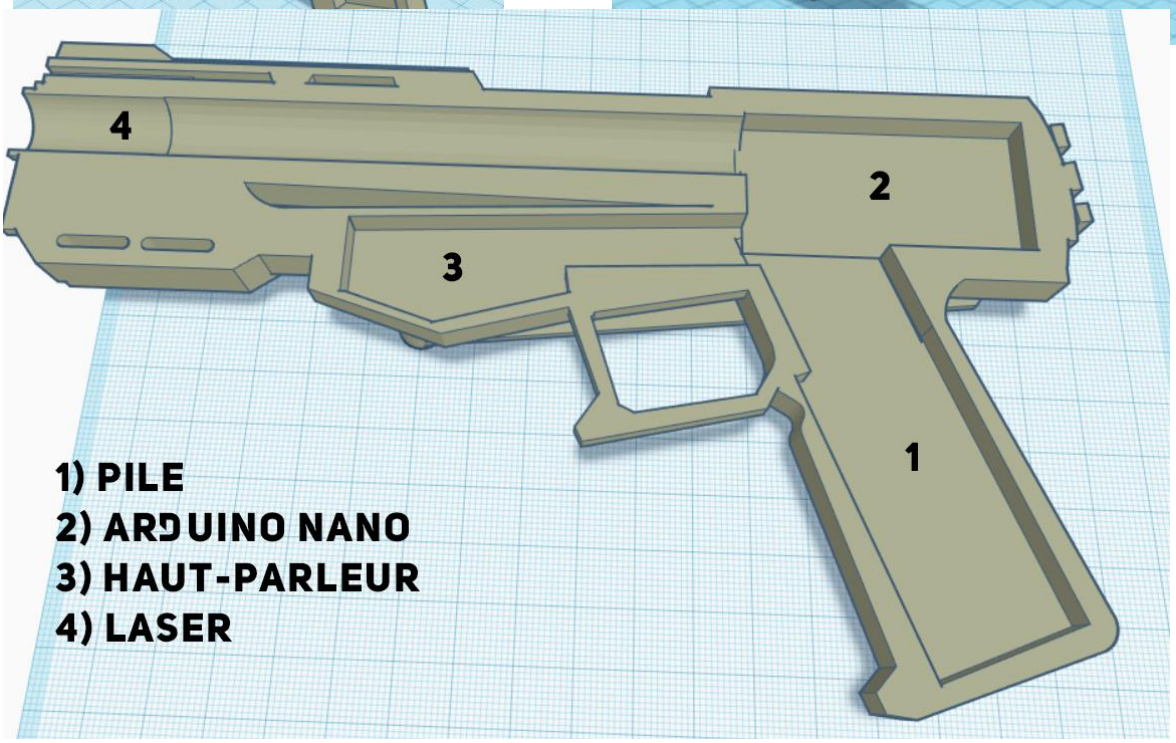
Vous aimez battre vos adversaires ? Ce mode est fait pour vous ! Battez et gagnez votre pari en ayant plus de points que votre adversaire. Il se pourrait que vous ayez une bière gratuite par votre adversaire… Comment cela se déroule ? Eh bien, 2 couleurs seront prises : **le rouge, le 1^{1er} joueur**, **le bleu, le 2^{e} joueurs**. Vous devrez touchez les cibles ayant la couleur assignée. **Attention !** Si vous touchez la cible de votre adversaire, cela lui rapportera 1 point !

Le fusil

Le fusil **sans fil** sera créé à l'aide d'un logiciel 3D et ensuite imprimé à l'aide d'une imprimante 3D. Dans celui-ci, nous y retrouverons notamment un **laser** qui va permettre de toucher la cible, un **bouton** qui permettra le rechargement du fusil et un **système de bruitage** lorsque le fusil tire et lorsqu'il touche la cible, un **microcontrôleur** y sera intégré afin de gérer le tout ainsi qu'une **pile** pour alimenter tout le matériel.

Style du fusil

J'aimerais utiliser un style de fusil assez « futuriste » ce pourquoi j'aimerais ajouter un design assez moderne.

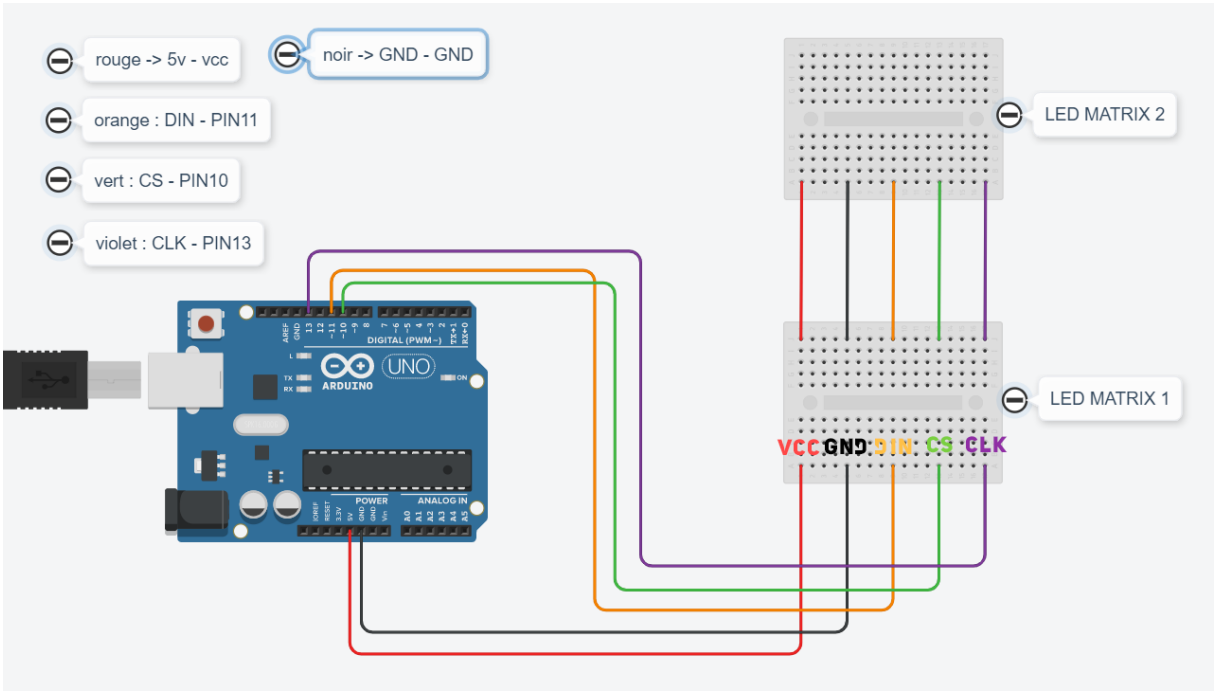


Le matériel

Désignation	Qte	Utilité	Prix/U (€)	Lien
Arduino Nano	2	Gestion du fusil.	3,45	Ebay
Arduino Uno	1	Gestion des cibles.	/	/
Module audio MP3 (x5)	1	Bruitage du fusil.	7,41	Amazon
Micro SDHC 8GO	2	Stockage fichier musicaux.	6	Big Tower
Module capteur laser (x10)	1	Cible du fusil (récepteur).	10,64 (Avec FDP)	Banggood
Module laser	2	Viseur du fusil (émetteur).	3,30	Ebay
Haut-Parleur (x5)	1	Bruitage des différents sons du fusil.	2,97 (+2,11 FDP)	AliExpress
Amplificateur (x5)	1	Augmenter le son des haut-parleurs	5,85	Amazon
BP LED POWER Rouge 5V (12mm)	2	Bouton qui sert au démarrage du fusil.	2,74 (+3,74 FDP)	Ebay
BP LED Jaune 5V (12mm)	2	Bouton qui sert au rechargement du fusil.	3,06 (+3,32 FDP)	Ebay
Anneau LED 8Bits	10	Anneau qui permet de savoir sur quelle cible tirée ainsi que son type	0,71 (+3,02 FDP)	AliExpress
Écran à matrice à points led	2	Affichage des scores.	9,89	Amazon
Pile 9V	2	Permet d'alimenter les fusils laser.	/	/
TOTAL : 103,04€				
Il se peut que cette liste change au fil du temps. (Frais de port compris pour tous les prix ci-dessus)				

Matrix LED

Câblage matrice led

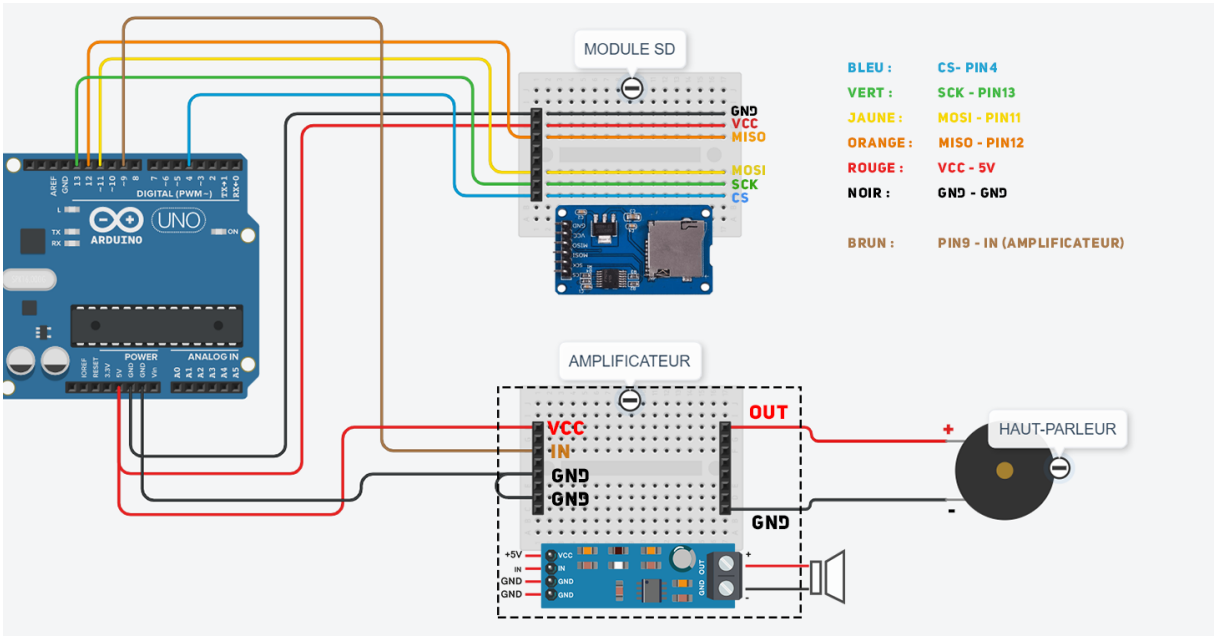


Code matrice led

```
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>

const uint16_t WAIT_TIME = 1000;

// Define the number of devices we have in the chain and the hardware interfac
e
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 8

#define CLK_PIN   13
#define DATA_PIN  11
#define CS_PIN    10

// Hardware SPI connection
MD_Parola P = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);
// Arbitrary output pins
// MD_Parola P = MD_Parola(HARDWARE_TYPE, DATA_PIN, CLK_PIN, CS_PIN, MAX_DEVIC
ES);

void setup(void)
{
    P.begin();
}

void loop(void)
{
    P.print("Hello");
    delay(WAIT_TIME);
    P.print(1234, DEC);
    delay(WAIT_TIME);
    P.print(1234, HEX);
    delay(WAIT_TIME);
    P.print(12.5);      // float not supported by Arduino Print class
    delay(WAIT_TIME);
    P.print(9876l);
    delay(WAIT_TIME);
    P.println("end");   // only get the /r/n characters - avoid using println
    delay(WAIT_TIME);
    P.write('A');
    delay(WAIT_TIME);
}
```

Module son

Câblage module son



Code module son

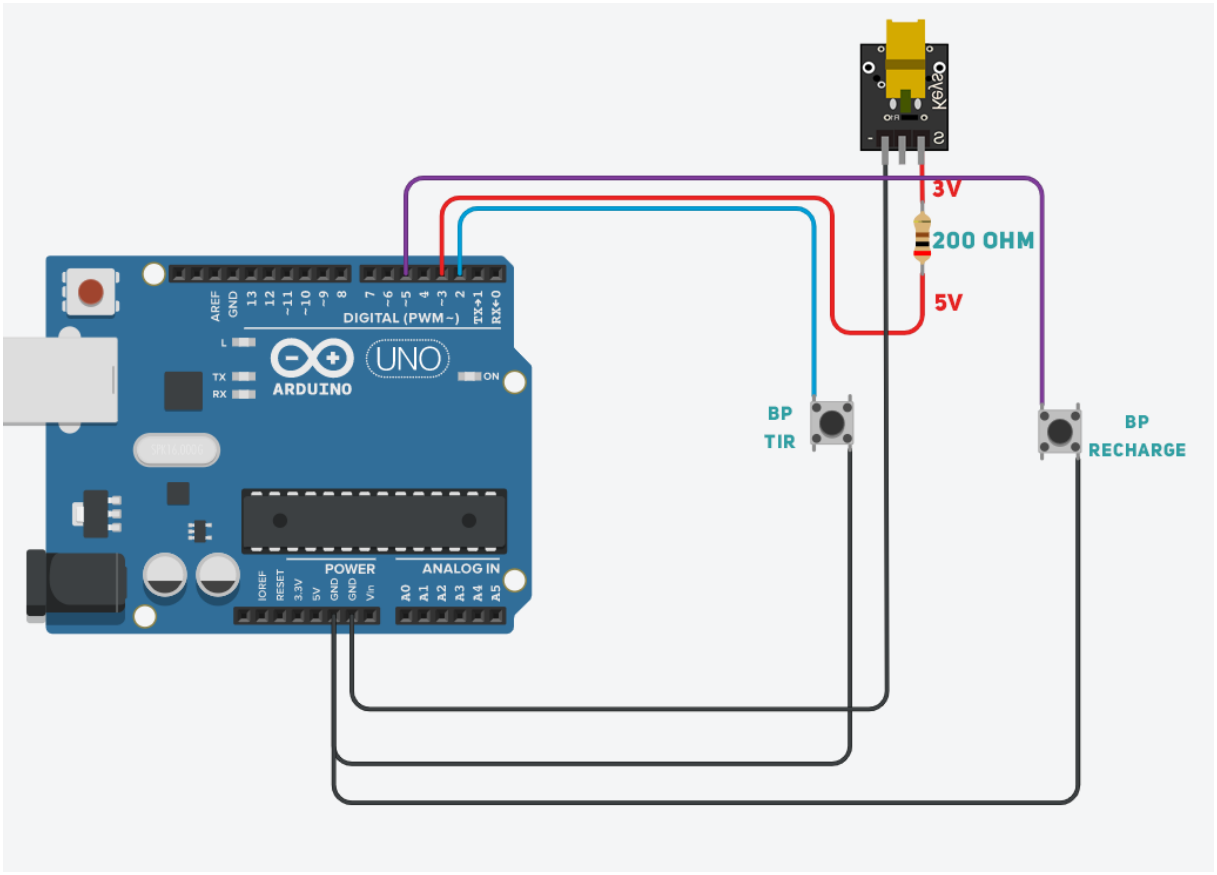
Attention, les fichiers audios du module SD devront être convertis en résolution **8 bits** ainsi qu'un taux d'échantillonnage de **16.000 Hz** et un canal audio en **mono** pour qu'il n'y ai pas de brouillage audio.

Convertisseur en ligne

```
#include <SD.h>                      // need to include the SD library
//#define SD_ChipSelectPin 53  //example uses hardware SS pin 53 on Mega2560
#define SD_ChipSelectPin 4  //using digital pin 4 on arduino nano 328, can use
 other pins
#include <TMRpcm.h>           //  also need to include this library...
#include <SPI.h>

TMRpcm tmrpcm;   // create an object for use in this sketch

void setup()
{
    tmrpcm.speakerPin = 9; //5,6,11 or 46 on Mega, 9 on Uno, Nano, etc

    Serial.begin(9600);
    if (!SD.begin(SD_ChipSelectPin))
    {  // see if the card is present and can be initialized:
        Serial.println("SD fail");
        return;   // don't do anything more if not
    }
    tmrpcm.play("music.wav"); //the sound file "music" will play each time the
 arduino powers up, or is reset
}

void loop()
{
    if(Serial.available())
    {
        if(Serial.read() == 'p')
        { //send the letter p over the serial monitor to start playback
        tmrpcm.play("music.wav");
        }
    }
}
```

Module laser

Câblage



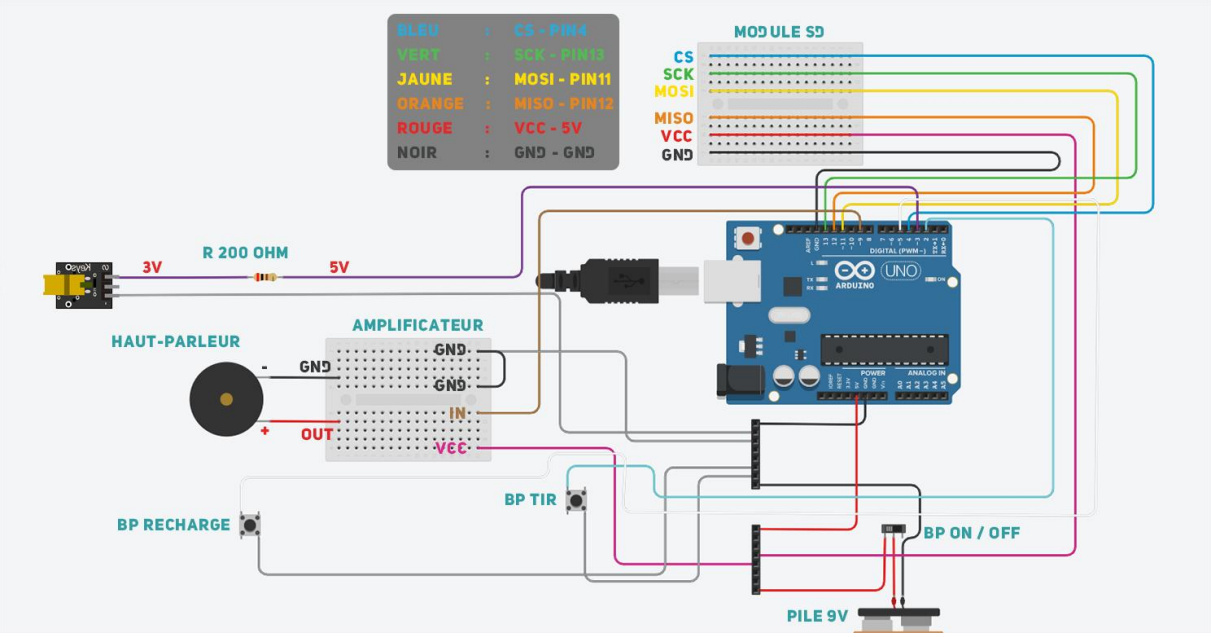
Code

```
int buttonShoot = 2;
int buttonReload = 5;
int laser = 3;

long timer;
boolean isPressed = false;
byte i = 0;

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

    pinMode(buttonShoot, INPUT_PULLUP);
    pinMode(buttonReload, INPUT_PULLUP);
    pinMode(laser, OUTPUT);
    digitalWrite(laser, LOW);
}

void loop()
{
    if(i < 7)
    {
        if(!digitalRead(buttonShoot))
        {
            if(!isPressed)
            {
                i++;
                isPressed = true;
                digitalWrite(laser, HIGH);
                timer = millis();
            }
        }
    }
    else if(i >= 7)
    {
        if(!digitalRead(buttonReload))
        {
            i = 0;
            delay(2000);
        }
    }

    if(digitalRead(buttonShoot))
    {
        if(isPressed)
```

```
        {
            isPressed = false;
            digitalWrite(laser, LOW);
        }
    }

    if((millis() - timer) > 3000 && isPressed)
    {
        digitalWrite(laser, LOW);
    }
}
```

Fusil (complet)

Câblage



Code

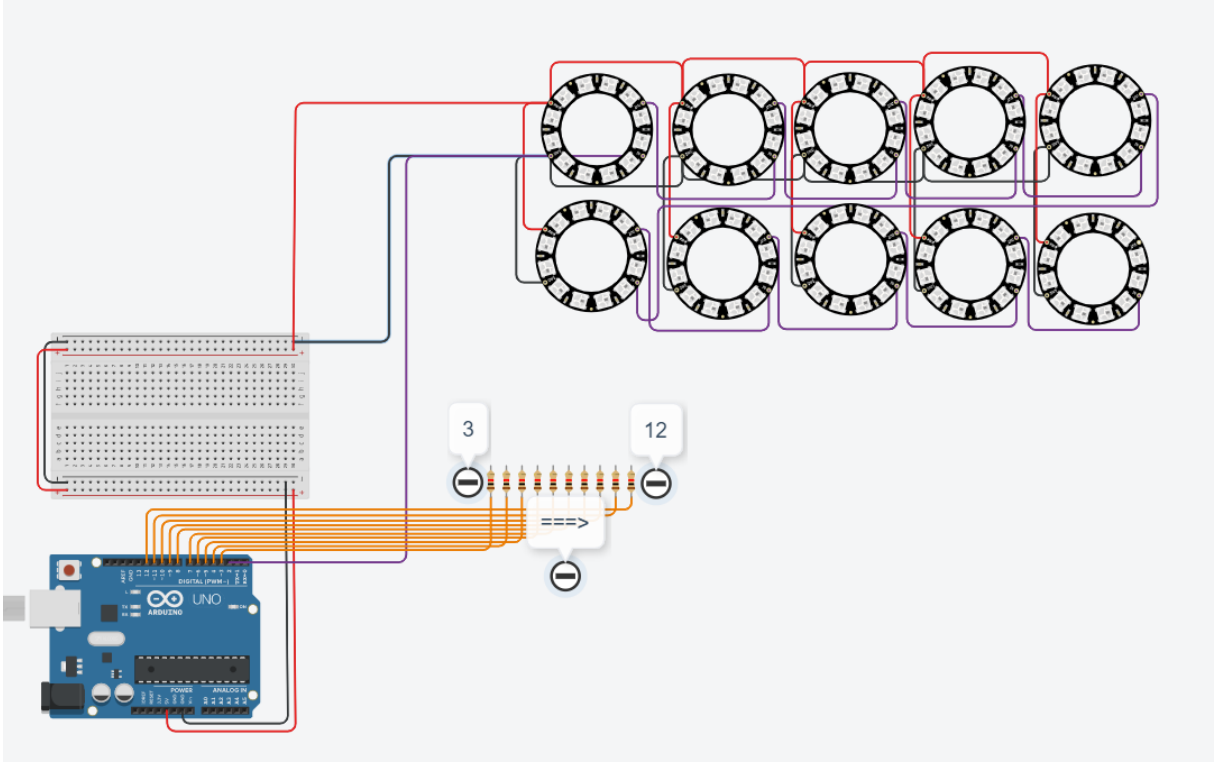
```
/*************************************************************/
/*                                                           */
/*      Projet  : Laser Game                                 */
/*      Date    : 24/02/2020                                 */
/*      Par     : Skytech                                    */
/*      Twitter : @Skytechh                                  */
/*      Github  : @Skytech61                                 */
/*                                                           */
/*    /!\ Toute réutilisation de ce code est interdit.       */
/*       Merci de respecter les droits d'auteur              */
/*                                                           */
/*************************************************************/

// ----------------------------------------------------------|
//
//                    INCLUDE LIBRARY
//
//----------------------------------------------------------|

// Song
#include <SD.h>
#include <TMRpcm.h>
#include <SPI.h>
TMRpcm tmrpcm;

// ----------------------------------------------------------|
//
//                      PIN SETUP
//
//----------------------------------------------------------|
#define SD_ChipSelectPin 4
int speakers = 9;
int buttonShoot = 2;
int buttonReload = 5;
int laser = 3;

// ----------------------------------------------------------|
//
//                      GAME CONFIG
//
//----------------------------------------------------------|
int bullets = 7; // bullet +1 (8 bullets)
int reloadTimer = 2000; // Time to reload gun
int pressTimer = 3000; // Time you can press shoot button
```

```
// ---------------------------------------------------------|
//
//                    VARIABLE DECLARATION
//
//----------------------------------------------------------|
long timer;
boolean isPressed = false;
byte i = 0;

// ---------------------------------------------------------|
//
//                          START CODE
//
//----------------------------------------------------------|
void setup()
{
    tmrpcm.speakerPin = speakers; // speaker pin;

    Serial.begin(9600);
    if (!SD.begin(SD_ChipSelectPin))
    {
        Serial.println("SD fail");
        return;
    }
    tmrpcm.play("start.wav");

    pinMode(buttonShoot, INPUT_PULLUP);
    pinMode(buttonReload, INPUT_PULLUP);
    pinMode(laser, OUTPUT);
    digitalWrite(laser, LOW);
}

void loop()
{
    if(i < bullets +1)
    {
        if(!digitalRead(buttonShoot))
        {
            if(!isPressed)
            {
                tmrpcm.play("shoot1.wav");
                i++;
                isPressed = true;
                digitalWrite(laser, HIGH);
                timer = millis();
            }
        }
    }
    else if(i >= bullets +1)
```

```
    {
        if(!digitalRead(buttonReload))
        {
            tmrpcm.play("reload.wav");
            i = 0;
            delay(reloadTimer);
        }
    }

    if(digitalRead(buttonShoot))
    {
        if(isPressed)
        {
            isPressed = false;
            digitalWrite(laser, LOW);
        }
    }

    if((millis() - timer) > pressTimer && isPressed)
    {
        digitalWrite(laser, LOW);
    }
}
```

Cibles (complet)

Câblage



Code (pas terminé)

```
/*********************************************************/
/*                                                       */
/*      Projet  : Laser Game                             */
/*      Date    : 24/02/2020                             */
/*      Par     : Skytech                                */
/*      Twitter : @Skytechh                              */
/*      Github  : @Skytech61                             */
/*                                                       */
/*    /!\ Toute réutilisation de ce code est interdit.   */
/*       Merci de respecter les droits d'auteur          */
/*                                                       */
/*********************************************************/

// ------------------------------------------------------|
//
//                    INCLUDE LIBRARY
//
//-------------------------------------------------------|

// NeoPixel ring
#include <Adafruit_NeoPixel.h>

// Matrix led
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>

// ------------------------------------------------------|
//
//                      PIN SETUP
//
//-------------------------------------------------------|
char soloButton = A0;
char multiButton = A1;
int restartButton = 13;
int ringsPin = 2;

// Matrix led pins
#define CLK_PIN   A2
#define DATA_PIN  A3
#define CS_PIN    A4

// ------------------------------------------------------|
//
//                     GAME CONFIG
```

```
//
//-------------------------------------------------------|
int turns = 3; // Number of turns
int waitingTime = 5000; // Waiting time to show NeoPixel ring
int pixelNumber = 12; // Number of pixels of NeoPixel ring

// -------------------------------------------------------|
//
//                   VARIABLE DECLARATION
//
//-------------------------------------------------------|

// Define the number of devices we have in the chain and the hardware interfac
e
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 8 // 4x2 matrix*/

// Hardware SPI connection
MD_Parola matrix = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);

// Declare NeoPixel strip object
Adafruit_NeoPixel strip(120, 2, NEO_GRB + NEO_KHZ800);

// NeoPixel colors
uint32_t red = strip.Color(255, 0, 0); // 20 points
uint32_t yellow = strip.Color(255, 255, 0); // 10 points
uint32_t blue = strip.Color(0, 0, 255); // 5 points
uint32_t green = strip.Color(0, 255, 0); // 1 points

// NeoPixel array for random
uint32_t colors[] = {red, yellow, blue, green};

// Point's color
byte scoreData[] = {1, 5, 10, 20};

byte i = 1;
byte score = 0;

// Declare any variable
long timer;
int color;
int cible;

boolean print = false;
boolean touch = false;

// -------------------------------------------------------|
//
//                          START CODE
```

```
//
//-------------------------------------------------------|
void setup()
{
    // Pin setup
    pinMode(soloButton, INPUT_PULLUP);
    pinMode(multiButton, INPUT_PULLUP);
    pinMode(restartButton, INPUT_PULLUP);
    pinMode(ringsPin, OUTPUT);

    // Initialise
    matrix.begin();
    strip.begin();
    strip.show();
    Serial.begin(9600);
    randomSeed(analogRead(0));
}

void loop()
{
    if(!digitalRead(restartButton))
    {
        Serial.println("isPressed");
        delay(1000);
    }
    else
    {
        Serial.println("notPressed");
        delay(1000);
    }

        if(i <= turns) // Number of turns
        {
            //Serial.println(i);
            if(!print)
            {
                // Generate random seed
                cible = random(0, 5);
                color = random(0, 4);

                // Show random led with random color during x second
                strip.fill(colors[color], (cible*pixelNumber), pixelNumber);
                strip.show();

                timer = millis();
                print = true;
                //Serial.println("cible allume: " + String(cible, DEC) + " cou
leurs: " + String(color, DEC));
            }
```

```
            else
            {
                if(millis() - timer >= waitingTime or touch)
                {
                    // After color clearing wait x second
                    strip.clear();
                    strip.show();
                    i++;

                    print = false;
                    touch = false;
                    delay(1000);
                }
                else if(digitalRead(cible+3)) // If targets are hit add points
                {
                    touch = true;
                    score += scoreData[color];
                }
            }
        }
        else // Game ended
        {
            //matrix.println("Score : " + String(score, DEC));
            //Serial.println("score: " + String(score, DEC));
            delay(4000);
        }
}
```

Amélioration possible

Si j'avais eu plus de temps, j'aurais ajouté ses différents éléments aux fusils : un **moteur** qui améliorerait l'expérience utilisateur en y ajoutant du dynamisme, un petit **écran LCD** afin de voir le nombre de balles restant. De plus, si j'avais eu plus de budget, j'aurais intégré un module rechargeable au fusil, ce qui aurais permis de ne plus consommés inutilement des piles 9V.

Enfin, si j'avais eu plus de temps et de budget, un module de communication (Bluetooth ou WI-FI) entre les fusils et le microcontrôleur des cibles aurait été ajoutés afin de pouvoir assigner les joueurs 1, 2 aux fusils laser.

Travail réalisé

Cibles :

- Construction du câblage des NéoPixels
- Développement du code des NéoPixels
- Construction du câblage des matrices LED
- Développement du code des matrices LED en relation avec les NéoPixels

Fusil :

- Construction 3D des fusils
- Construction du câble du module son avec amplificateur
- Développement du module audio
- Test du module audio
- Construction du câblage des Lasers
- Développement du bouton de recharge
- Développement du bouton de tire

Mise en œuvre réelle du développement ainsi que du montage des cibles sur une planche de bois ainsi que le câblage de tout le matériel utilisé + mise en couleur.

Les rapports remis vous donneront de manière plus précise les tâches effectués.