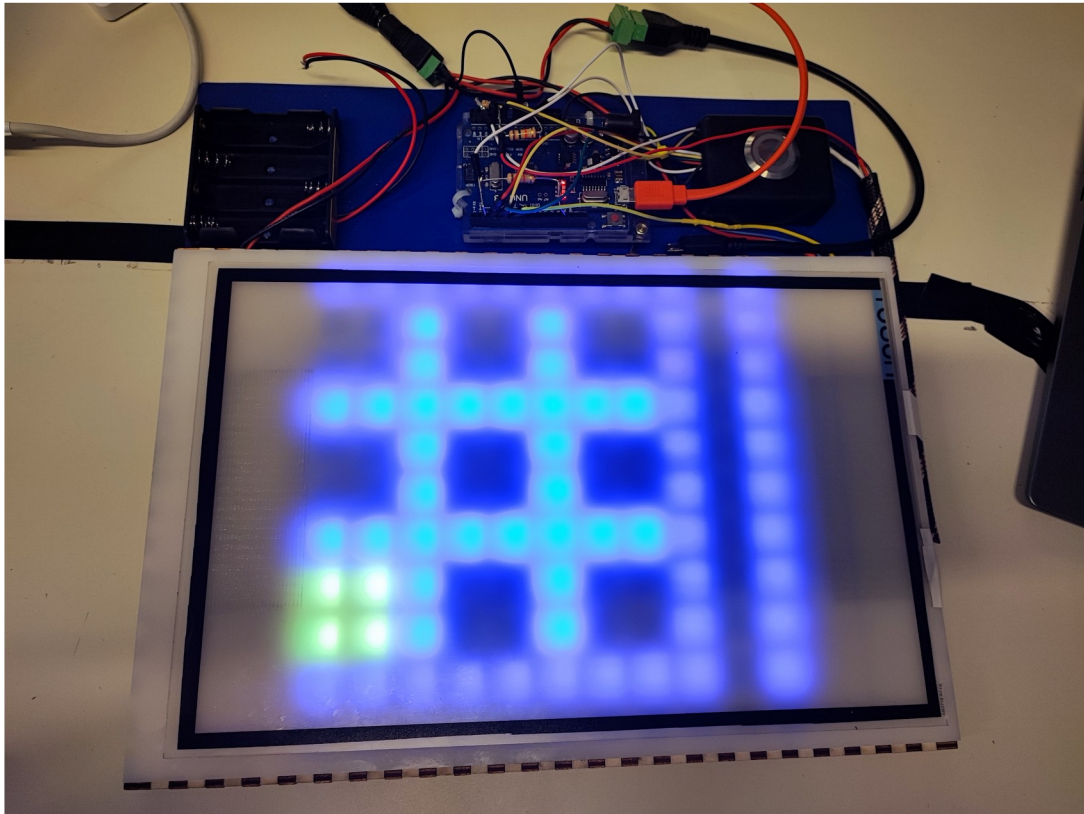


Rapport Séance 3 :

Nous avons fait en sorte que le dessin de la grille et l'emplacement du score s'affiche sur l'écran. Voici le résultat :



Nous avons ensuite fait le programme qui nous récupère les coordonnées de l'appui tactile et associé chaque case de la grille à des coordonnées minimales en x et en y et les quatre LEDs qui la composent.

Voici les programmes ;

```
#include <Adafruit_NeoPixel.h>
#include <stdint.h>
#include "TouchScreen.h"

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
int X=120;

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

#define YP A0 // must be an analog pin, use "An" notation!
#define XM A3 // must be an analog pin, use "An" notation!
#define YM A1 // can be a digital pin
#define XP A2 // can be a digital pin

int grille[28] = {13,16,27,30,33,34,35,36,37,38,39,40,49,52,57,60,69,70,71,72,73,74,75,76,79,82,93,96};

int grille2[36] = {0,2,3,4,5,6,7,8,9,10,19,21,24,22,41,43,44,46,63,65,66,68,85,87,88,90,99,107,106,105,104,103,102,101,100,109};
```

```
int case1[8] = {205,330, 440,535, 18,17,26,25};
int case2[8] = {205,330, 590,695, 14,15,29,28};
int case3[8] = {205,330, 770,840, 12,11,31,32};
int case4[8] = {450,575, 440,535, 47,48,61,62};
int case5[8] = {450,575, 590,695, 50,51,58,59};
int case6[8] = {450,575, 770,840, 53,54,55,56};
int case7[8] = {690,835, 440,535, 83,84,91,92};
int case8[8] = {690,835, 590,695, 80,81,94,95};
int case9[8] = {690,835, 770,840, 77,78,97,98};
```

```
Adafruit NeoPixel strip = Adafruit NeoPixel(X, 6, NEO_GRB + NEO_KHZ800);
void setup() {
    strip.begin();
    strip.show(); // Initialise toute les led à 'off'
    Serial.begin(9600);
}

void loop() {
    for(int i = 0; i < 28; i++) { // On fait une boucle pour définir la couleur de chaque led
        // setPixelColor(n° de led, Rouge, Vert, Bleu)
        strip.setPixelColor(grille[i], 0,0,255);
    } // on affiche // On fait une boucle pour définir la couleur de chaque led
        // setPixelColor(n° de led, Rouge, Vert, Bleu)
    for(int i = 0; i < 36; i++) { // On fait une boucle pour définir la couleur de chaque led
        // setPixelColor(n° de led, Rouge, Vert, Bleu)
        strip.setPixelColor(grille2[i], 0,0,50);
    } // on affiche // On fait une boucle pour définir la couleur de chaque led
    strip.setPixelColor(77, 255, 255, 0);
    strip.setPixelColor(78, 255, 255, 0);
    strip.setPixelColor(97, 255, 255, 0);
    strip.setPixelColor(98, 255, 255, 0);
    strip.show(); // on affiche
}
```

```
// Touch screen library with X Y and Z (pressure) readings as well
// as oversampling to avoid 'bouncing'
// This demo code returns raw readings, public domain
```

```
#include <stdint.h>
#include "TouchScreen.h"
```

```
#define YP A0 // must be an analog pin, use "An" notation!
#define XM A3 // must be an analog pin, use "An" notation!
#define YM A1 // can be a digital pin
#define XP A2 // can be a digital pin
```

```
// For better pressure precision, we need to know the resistance
// between X+ and X- Use any multimeter to read it
// For the one we're using, its 300 ohms across the X plate
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
```

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

```
void loop(void) {
    // a point object holds x y and z coordinates
    TSPoint p = ts.getPoint();

    // we have some minimum pressure we consider 'valid'
    // pressure of 0 means no pressing!
    if (p.z > 0) {
```

```
Serial.print("X = "); Serial.print(p.x);  
Serial.print("\tY = "); Serial.print(p.y);  
Serial.print("\tPressure = "); Serial.println(p.z);
```

```
}
```