**Rendu séance n°7**

Aujourd’hui, nous avons finalement réussi à finir de coder notre jeu avec le tactile de l’écran qui fonctionne bien. Cependant il nous reste encore un bug avec la case 3 qui s'allume lorsqu’on appuie sur 3 des 9 cases (pour l’instant nous n’avons aucune idée d’où provient ce problème).

Nous avons aussi codé le bouton poussoir. Il permet de commencer une partie et d’éteindre toutes les leds lorsque nous ne voulons plus jouer.

Notre programme :

#include <Adafruit\_NeoPixel.h> //Allumage des LEDs

#include "TouchScreen.h"       // Tactile : position de l'appui et pression

//nombre de LEDs

int X=120;

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(X, 6, NEO\_GRB + NEO\_KHZ800);

// Parameter 32 = 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)

//Entrées de la plaque tactile

#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

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 0);

const int bouton = 2;

const int rougeB = 3;

const int verteB = 4;

const int bleueB = 5;

int game = 1;

int val = 0;

int ancien\_val = 1;

//-----------------------------------VARIABLES----------------------------------------------------------------------------------------------------------------------//

int joueur = -1;   // 1 = J1 (jaune) et -1 = J2 (rouge)

//numéro des LEDs à allumer pour l'affichage de l'écran de victoire

int winOrange[19] = {54,55,76,78,74,80,72,59,50, 48,61,70,83,62,68,67,64,45,86};

int winJaune[7] = {49,60,71,82,44,65,87};

int winRouge[22] = {32,31,30,29,28,27,26,25,24,23,22, 99,100,101,102,103,104,105,106,107,108,109};

//certaines LEDs seront à éteindre après l'affichage de l'écran de victoire

int winDelete[6] = {45,64,67,86,23,108};

int tieOrange[17] ={54, 53, 52, 56,75,78,97,48,47,46,61,69,70,83,92,91,90};

int tieJaune[13] = {49,50,51,59,72,81,93,94,95,44,65,66,88};

int casesAllumees[9] = {10, 10, 10, 10, 10, 10, 10, 10, 10};  // liste de toutes les cases allumées

int casesJ1[9] = {10, 10, 10, 10, 10, 10, 10, 10, 10};        // liste des cases allumées par le J1

int casesJ2[9] = {10, 10, 10, 10, 10, 10, 10, 10, 10};        // liste des cases allumées par le J2

int Score1[5] = {20,23,42,45,64};  //liste des LEDs où afficher le score du J1

int Score2[5] = {108,89,86,67,64};  //liste des LEDs où afficher le score du J2

int scoreJ1 = 0;  //enregistre le score du J1

int scoreJ2 = 0; //enregistre le score du J2

bool caseOk=true;  //permet de vérifier si une case n'est pas déjà allumée

//liste des LEDs qui constitue la grille/plateau du jeu (en orange)

int grille[65] = {0,1,2,3,4,5,6,7,8,9,10,19,21,24,22,41,43,44,46,63,65,66,68,85,87,88,90,99,107,106,105,104,103,102,101,100,109,13,16,27,30,33,34,35,36,37,38,39,40,49,52,57,60,69,70,71,72,73,74,75,76,79,82,93,96};

//pour chaque case allumable : {x\_min,x\_max, y\_min,y\_max, LEDs qui la constitue}

int cases[9][8] = {{500,600,  680,810, 18,17,26,25},

                  {340,430,  680,810, 14,15,29,28},

                  {180,290,  680,810, 12,11,31,32},

                  {500,600,  460,580, 47,48,61,62},

                  {340,430,  460,580, 50,51,58,59},

                  {180,290,  460,580, 53,54,55,56},

                  {500,600,  230,350, 83,84,91,92},

                  {340,430,  230,350, 80,81,94,95},

                  {180,290,  230,350, 77,78,97,98}};

//chaque combinaison de cases permettant de gagner

int combiWin[8][3]= {{0,1,2},

                  {3,4,5},

                  {6,7,8},

                  {0,3,6},

                  {1,4,7},

                  {2,5,8},

                  {2,4,6},

                  {0,4,8}};

//------------------------------------FONCTIONS/METHODES--------------------------------------------------------------------------------------------------------------//

//------------------ALLUMAGE DES LEDS/AFFICHAGE----------------//

//allume en début de manche la grille/plateau (en orange)

void affichage(){

  for(int i = 0; i < 65; i++ ) {

    strip.setPixelColor(grille[i], 255,69,0);

  }

  strip.show(); // on affiche

}

//allume les LEDs qui montre le score des joueurs

void afficheScore(){

  for (int j=0; j<scoreJ1; j++){

    strip.setPixelColor(Score1[j], 255, 255, 0);

  }

  for (int k=0; k<scoreJ2; k++){

    strip.setPixelColor(Score2[k], 255, 0, 0);

  }

  strip.show();

}

//permet d'éteindre les cases pour recommencer une manche

void reset(){

  for (int i=0; i<9; i++){

    for (int j=4; j<8; j++){

      strip.setPixelColor(cases[i][j], 0,0,0);

    }

  }

  strip.show();

}

//permet d'éteindre les LEDs affichant le score pour recommencer une partie

void resetScore(){

  for (int i = 0; i<5; i++){

    strip.setPixelColor(Score1[i], 0,0,0);

    strip.setPixelColor(Score2[i], 0,0,0);

  }

}

//affiche un écran de victoire quand une manche est gagnée

void affichageWin(){

  strip.clear();

  for (int i=0; i<19; i++){

    strip.setPixelColor(winOrange[i], 255,69,0);

  }

  for (int i=0; i<7; i++){

    strip.setPixelColor(winJaune[i], 255,255,0);

  }

  for (int i=0; i<22; i++){

    strip.setPixelColor(winRouge[i], 255,0,0);

  }

  strip.show();

  delay(3000);

  for (int i=0; i<6; i++){

    strip.setPixelColor(winDelete[i], 0,0,0);

  }

  affichage();

  afficheScore();

}

// affiche TIE ! lors d'une égalité

void affichageTie(){

  strip.clear();

  for (int i=0; i<17; i++){

    strip.setPixelColor(tieOrange[i], 255,69,0);

  }

  for (int i=0; i<13; i++){

    strip.setPixelColor(tieJaune[i], 255,255,0);

  }

  for (int i=0; i<22; i++){

    strip.setPixelColor(winRouge[i], 255,0,0);

  }

  strip.show();

  delay(3000);

  strip.setPixelColor(23, 0,0,0);

  strip.setPixelColor(108, 0,0,0);

  affichage();

  afficheScore();

}

//--------------------GESTION DES CASES-----------------------//

//prend une position en x et en y et renvoie l'index de la case correspondant à cette position

int indexCase(float x, float y){

  if (x>650 ||(y <679 && y>581)){

    return 10;

  }

  else{

  bool breakk = false;

  int index = 0;

  while (breakk == false){

    if (cases[index][0]<x && x<cases[index][1] && cases[index][2]<y && y<cases[index][3]){

      return index;

      breakk = true;

    }

    index += 1;

  }

  }

}

//vérifie si une case est jouable, c'est-à-dire qu'elle n'est pas allumées

void verifyCase(int index){

  for (int i =0; i<9; i++){

    if (index == casesAllumees[i] || index>8){

      caseOk = false;

    }

  }

}

// remplace un 10 dans les listes des cases allumées par la case qui vient d'être jouée.

void caseJouee(int index){

  int index10 = 0;

  //on remplace d'abord dans la liste de toutes les cases allumées.

  for (int i=0; i<9; i++){

    if (casesAllumees[i]>=10){

      index10=i;

    }

  }

  casesAllumees[index10] = index;

  //ensuite on remplace dans la liste du joueur qui vient de jouer

  if (joueur == 1){

    casesJ1[index10]=index;

  }

  else{

    casesJ2[index10]=index;

  }

}

//remplit les listes des cases jouées par des 10 pour recommencer la partie.

void cases10(){

  for (int i=0; i<9; i++){

    casesAllumees[i] =10;

    casesJ1[i] =10;

    casesJ2[i]=10;

  }

}

void verifyList(){

  for (int i =0; i<9; i++){

    if (casesAllumees[i]>10){

      casesAllumees[i]=10;

    }

  }

}

//--------------------FIN DE PARTIE----------------------//

//vérifie si un joueur a gagné ou si toutes les cases sont allumées sans gagnant : renvoie true sinon false.

bool verifyEnd(){

  // On vérifie si un joueur a gagné : on parcourt la liste des combinaisons gagnantes et on vérifie si une d'entres elles apparaît dans la liste des cases allumées par un joueur.

  for (int i=0; i<8; i++){

    int ok =0;

    for (int j=0; j<3; j++){

      for (int k=0; k<9; k++){

        if (joueur == 1 && combiWin[i][j]==casesJ1[k]){

          ok += 1;

          if (ok==3){

            cases10();

            scoreJ1 += 1;

            affichageWin();

            if (scoreJ1 == 5){

              joueur = -1;

              scoreJ1 = 0;

              scoreJ2 = 0;

              resetScore();

            }

            return true;

          }

        }

        else if (joueur == -1 && combiWin[i][j]==casesJ2[k]){

          ok += 1;

          if (ok==3){

            cases10();

            scoreJ2 += 1;

            affichageWin();

            if (scoreJ2 == 5){

              joueur = -1;

              scoreJ2 = 0;

              scoreJ1 = 0;

              resetScore();

            }

            return true;

          }

        }

      }

    }

  }

  // on vérifie ensuite si toutes les cases sont allumées : dans ce cas personne ne peut gagner : on met fin à la partie

  int count = 0;

  for (int h = 0; h<9; h++){

    if (casesAllumees[h]<10){

      count += 1;

    }

  }

  if (count ==9){

    cases10();

    affichageTie();

    return true;

  }

  // on return false : la partie n'est pas finie.

  return false;

}

void off(){

  cases10();

  reset();

  scoreJ1 = 0;

  scoreJ2 = 0;

  resetScore();

  for (int i=0; i<65; i++){

    strip.setPixelColor(grille[i], 0,0,0);

  }

}

//--------------------------------------POSITION-----------------------------------------------------------------------------------------------------------------------//

int posY(){

  pinMode(17, OUTPUT);

  pinMode(16, OUTPUT);

  pinMode(14, INPUT);

  pinMode(15, INPUT);

  digitalWrite(16, HIGH);

  digitalWrite(17, LOW);

  return analogRead(1);

}

int posX(){

  pinMode(17, INPUT);

  pinMode(16, INPUT);

  pinMode(14, OUTPUT);

  pinMode(15, OUTPUT);

  digitalWrite(14, HIGH);

  digitalWrite(15, LOW);

  return analogRead(3);

}

//--------------------------------------------------------SETUP--------------------------------------------------------------------------------------------------------//

void setup() {

  strip.begin();

  pinMode(bouton,INPUT);

  Serial.begin(9600);

  strip.show(); // Initialise toute les led à 'off'

}

//----------------------------------------------------------LOOP------------------------------------------------------------------------------------------------------//

void loop() {

  Serial.println(game);

  val = digitalRead(bouton);

  if ((val == LOW)&&(ancien\_val==HIGH)){

    game = 1-game;

    delay(20);

  }

  ancien\_val = val;

  if (game ==0){

    off();

  }

  else{

    caseOk = true;

    affichage();

    afficheScore();

    int x = posX();

    int y = posY();

    int index = indexCase(x,y);

    verifyCase(index);

    if (caseOk==true){

      joueur = -joueur;

      caseJouee(index);

      for (int j=4; j<8;j++){

        if (joueur == 1){

          strip.setPixelColor(cases[index][j], 255,255,0);

        }

        else{

          strip.setPixelColor(cases[index][j], 255,0,0);

        }

      }

    }

    if (verifyEnd()==true){

      reset();

    }

  }

}