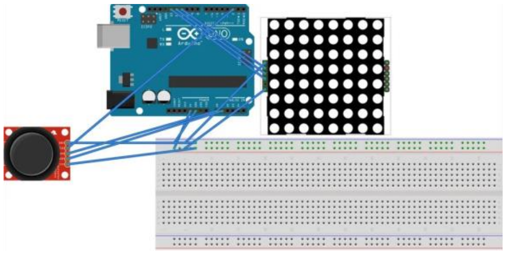
Juego Snake con Matriz Led 8x8



- ***Descripción del ejercicio:***

Este esta práctica crearemos el juego de serpiente portátil (snake), para esto aprenderemos a conectar los componentes más importantes la matriz LED y el control de mando (joystick).

- ***Materiales***

  - 1 Arduino UNO
  - 1 Módulo de joystick
  - 1 Matriz Led 8x8
  - 1 Placa de pruebas (Protoboard)
  - 1 Alambre para conexiones

- ***El código que usaremos será el siguiente***

```
#include "LedControlMS.h"
#define NBR_MTX 2
LedControl lc=LedControl(13,12,11, NBR_MTX);
String digits= "1234567890";
int digitCounter=0;
/* we always wait a bit between updates of the display */
unsigned long delaytime=300;
const int SW_pin = 2; // digital pin connected to switch output
const int X_pin = 0; // analog pin connected to X output
const int Y_pin = 1; // analog pin connected to Y output
int foodX = 4;
int foodY = 5;
int snakeX[100];
int snakeY [100];
int dir = 0;
int len = 1;
int length1 = 0;
int length2 = 0;
```

```
boolean play = false;
void setup() {
/* The MAX72XX is in power-saving mode on startup, we have to do a
wakeup call */ Serial.begin (9600);
pinMode(SW_pin, INPUT);
digitalWrite(SW_pin, HIGH);
digitCounter=0;
for (int i=0; i< NBR_MTX; i++){
lc.shutdown(i,false);
/* Set the brightness to a medium values */
lc.setIntensity(i,8);
/* and clear the display */
lc.clearDisplay(i);
}
snakeX[0] = 4;
snakeY[0] = 2;
snakeX[1] = 4;
snakeY[1] = 1;
lc.setLed(0,foodX,foodY,true);
lc.setLed(0,snakeX[0],snakeY[0],true);
delay(1000);
}
void loop() {
if (digitalRead(SW_pin)==0){
play = true;
}
if (play == true) {
lc.clearAll();
for (int i=len; i>0; i--) {
snakeX[i]=snakeX[i-1];
snakeY[i]=snakeY[i-1];
} if (dir == 0) {
if (analogRead(X_pin)>800) {
dir += 1;
snakeX[0] += 1;
} else if (analogRead(X_pin)<200) {
dir = 3;
snakeX[0] -= 1;
} else snakeY[0] += 1;
} else if (dir == 1) {
if (analogRead(Y_pin)>800) {
dir += 1;
snakeY[0] -= 1;
} else if (analogRead(Y_pin)<200) {
dir -= 1;
snakeY[0] += 1;
} else snakeX[0] += 1;
} else if (dir == 2) {
if (analogRead(X_pin)>800) {
dir -= 1;
snakeX[0] += 1;
} else if (analogRead(X_pin)<200) {
dir += 1;
snakeX[0] -= 1;
} else snakeY[0] -= 1;
} else if (dir == 3) {
if (analogRead(Y_pin)>800) {
dir -= 1;
snakeY[0] -= 1;
} else if (analogRead(Y_pin)<200) {
dir = 0;
snakeY[0] += 1;
} else snakeX[0] -= 1;
}
if (snakeX[0]>7){
snakeX[0] = 0;
}
if (snakeX[0]<0){
snakeX[0] = 7;
}
if (snakeY[0]>7){
snakeY[0] = 0;
}
if (snakeY[0]<0){
snakeY[0] = 7;
}
if (snakeX[0] == foodX and snakeY[0] == foodY) {
foodX=random(8);
foodY=random(8);
len+=1;
length1+=1;
}
if (length1==10){
length1=0;
length2+=1;
}
for (int i = 1; i< len; i++){
if(snakeX[0]==snakeX[i] and snakeY[0]==snakeY[i]){
gameOver();
}
}
lc.setLed(0,foodX,foodY,true);
for (int i = 0; i< len; i++){
lc.setLed(0,snakeX[i],snakeY[i],true);
}
delay(500);
}
}
void gameOver(){ lc.writeString(0,"Game Over");
delay(1000);
```

```
lc.writeString(0,"Score");
delay(1000);
lc.displayChar(0,length2);
delay(500);
lc.clearDisplay(0);
delay(200);
lc.displayChar(0,length1);
delay(1000);
softwareReset();
}
void softwareReset() {
foodX = 4;
foodY = 5;
dir = 0;
len = 1;
length1 = 0;
length2 = 0;
play = false;
snakeX[0] = 4;
snakeY[0] = 2;
snakeX[1] = 4;
snakeY[1] = 1;
lc.clearDisplay(0);
Serial.println("LED0: foodX foodY");
lc.setLed(0,foodX,foodY,true);
Serial.println("LED0: snakeX, snakeY");
lc.setLed(0,snakeX[0],snakeY[0],true);
delay(1000);
}
void scrollLeft(char ch){
int pos =lc.getCharArrayPosition(ch);
for (int scroll =0; scroll<6; scroll++) {
for (int i=scroll; i<6;i++) {
lc.setRow(0,i-scroll, alphabetBitmap[pos][i]);
}
delay(300);
lc.clearDisplay(0);
}
}
void scrollRight(char ch){
int pos =lc.getCharArrayPosition(ch);
for (int scroll =0; scroll<8; scroll++) {
for (int i=0; i<6;i++) {
if (scroll+i<8) lc.setRow(0, scroll+i, alphabetBitmap[pos][i]);
}
delay(300);
lc.clearDisplay(0);
}
}
```