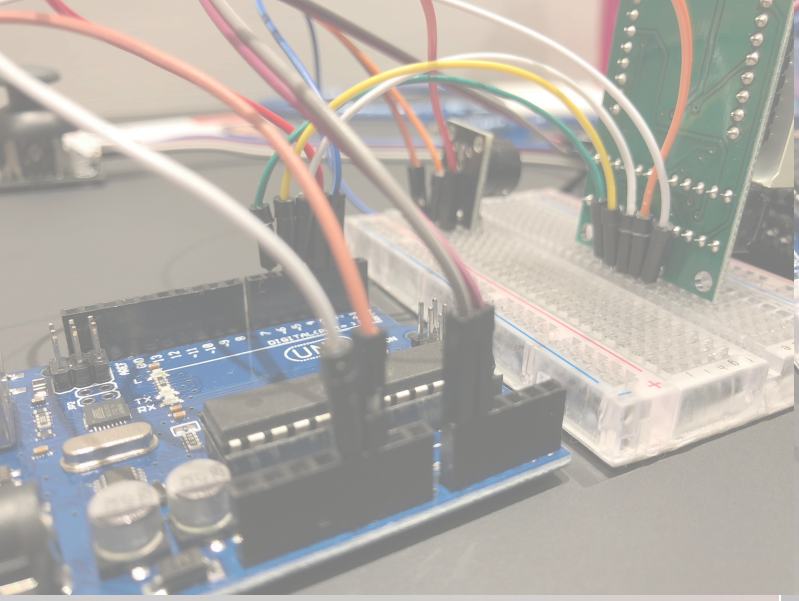
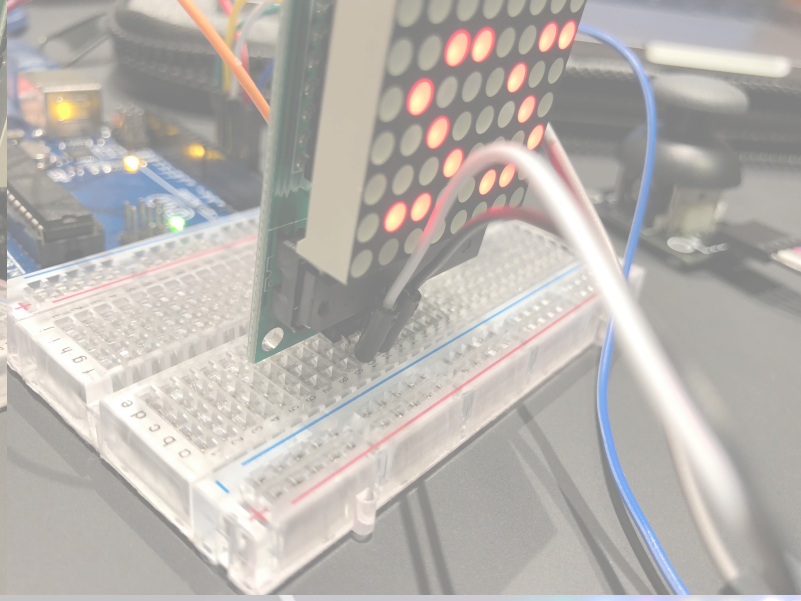
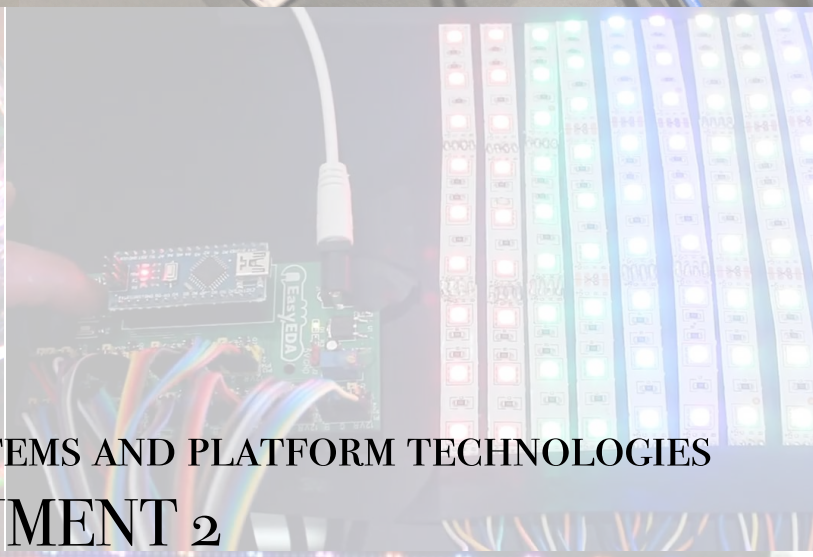
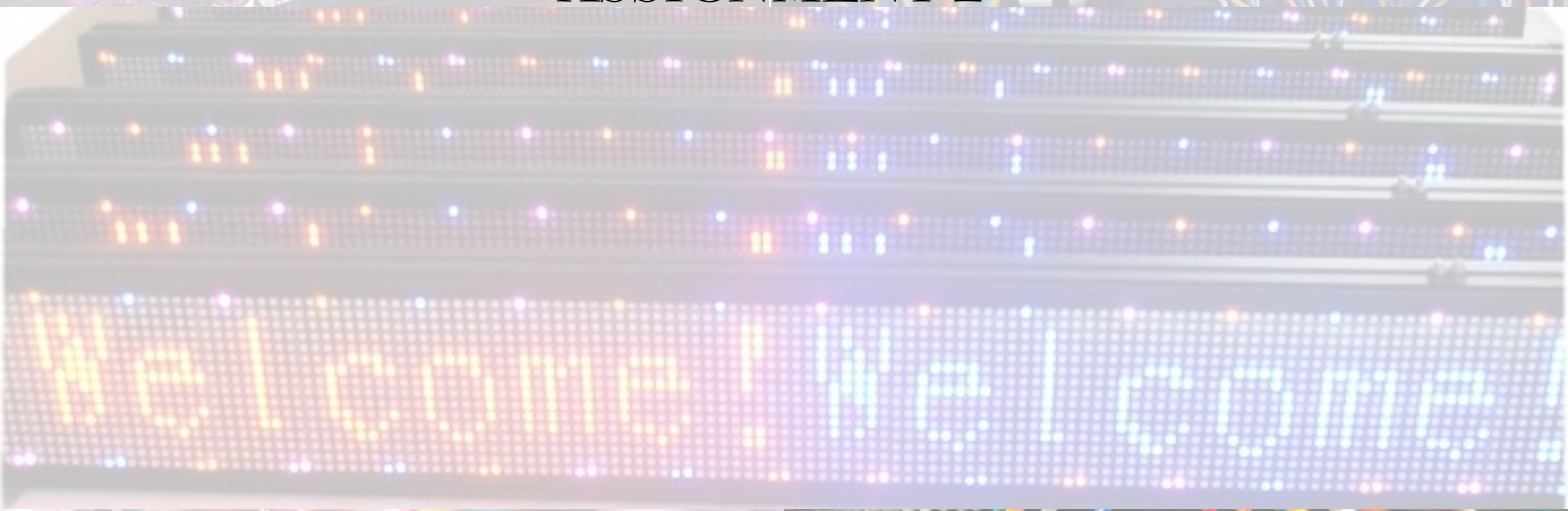
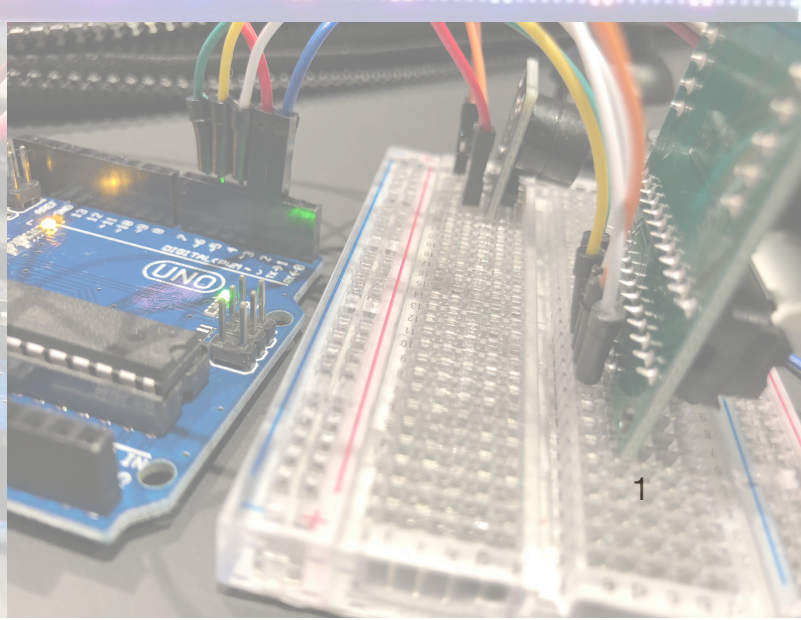





Introduction to computer systems and platform technologies

ASSIGNMENT 2



Milindi Kodikara -s3667779

Contents

Introduction

One of the most important developments in computing in recent years is the Internet of Things, in which many appliances and machines can be controlled by small micro processors which can be connected to the internet and thus controlled remotely or coordinated as part of a larger “intelligent” system. This is a small project that demonstrates the basic concepts of IoT devices using a micro-controller.

As a Software Engineering student at RMIT there is not much to our life than sitting in-front of a computer and writing code, therefore, when our lecturer informed us about the choice between doing the paper or the Arduino project I immediately set my heart on doing the Arduino project as it intrigued me immensely. As I’ve never build a project of my own, finding an innovative idea proved to be quite the challenge as my knowledge on building simple hardware was inadequate as well. Nevertheless, I practiced the given examples and surfed the internet for ideas on a regular basis.

Ever since I was a kid who was just beginning to get obsessed with a Nintendo Game Boy, I have always wondered what it would be like to create my very own game. Therefore, I took this amazing opportunity to make that simple dream into a reality by creating a game long forgotten by many, which would eventually leave everyone with a feeling of nostalgia.

Snake Game has been an all time favourite since the beginning of the usage of mobile phones. Initially it was used in black and white cellular phones. Then with the advancement of cellphones, this game went through various changes, now many graphical and colourful versions of this game are available.

The project I have demonstrated is a modified version of the Snake Game, with all its basic functionalities, which I have adapted from an off the shelf solution.

Components used

Arduino Uno
USB cable
Breadboard
Jumper Cables
8*8 Matrix Module
Buzzer Module
Joystick Module

Component descriptions

1.Arduino Uno

The Arduino UNO is a widely used open-source micro-controller board based on the ATmega328P micro-controller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

2.Matrix Module

A dot-matrix display is a display device used to display information with limited resolution. The display consists of a dot-matrix of lights or mechanical indicators arranged in a rectangular configuration such that by switching on or off selected lights, text or graphics can be displayed. A dot-matrix controller converts instructions from a processor into signals which turns on or off lights in the matrix so that the required display is produced.

3.Buzzer Module

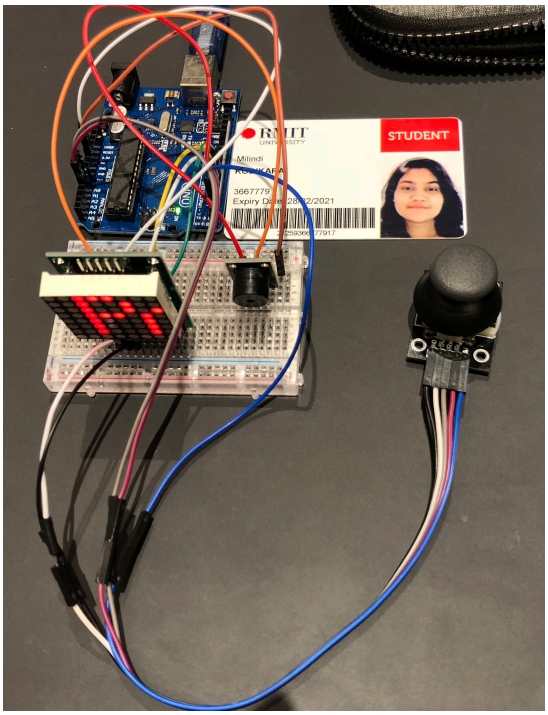
A buzzer is an audio signalling device which uses DC power supply. As a type of electronic buzzer with integrated structure. Buzzers can be categorised as active and passive buzzers

4.Joystick Module

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling.

Circuit Diagram

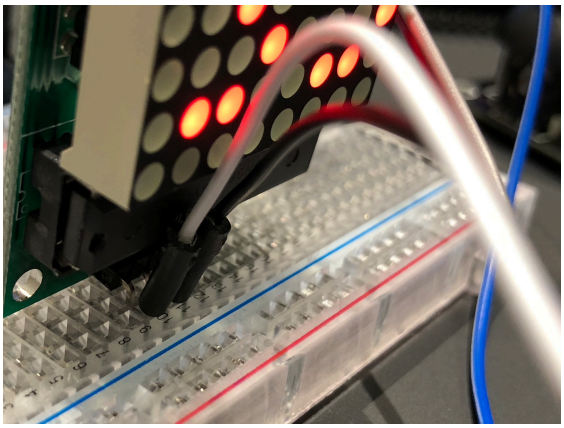
Uno	Matrix	Buzzer	Joystick
D2			SW
D3		S	
D4	DIN		
D5	CS		
D6	CLK		
A0			VRx
A1			VRy
5V	VCC		+5V
GND	GND	-	GND

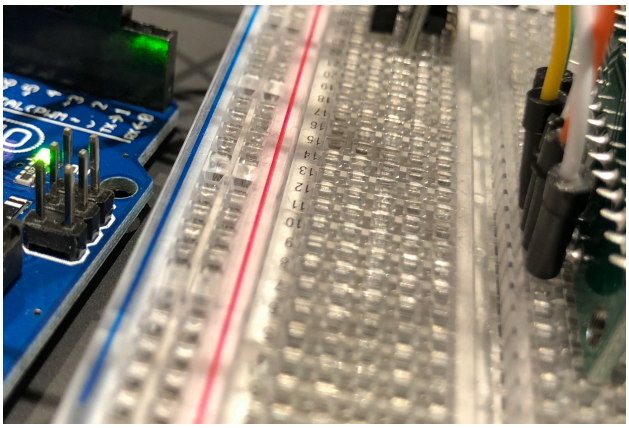


Working

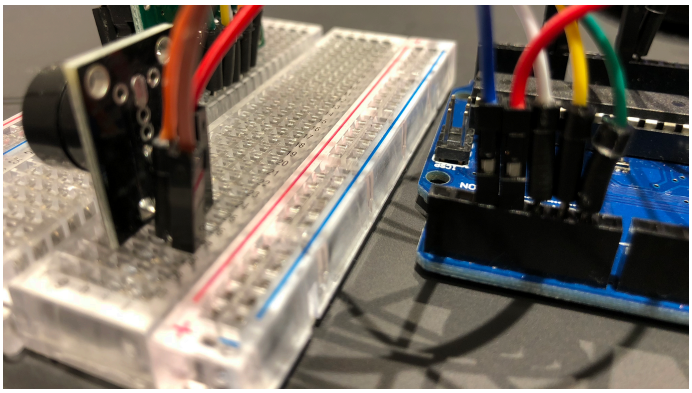
The project is set up in the following method:

1.Plug in the jumper wires as depicted below on the breadboard along with the matrix module.

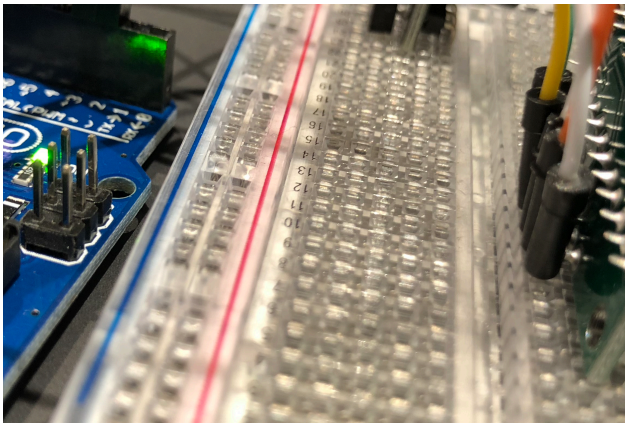


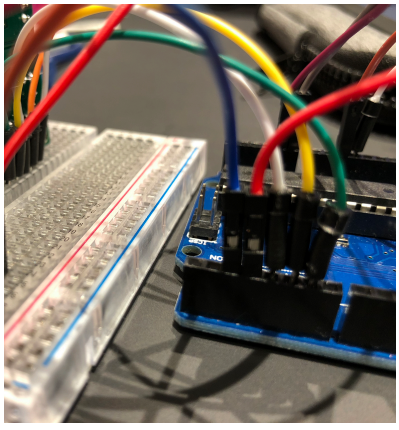


2.Insert the buzzer module to the right of the jumper wires implemented.

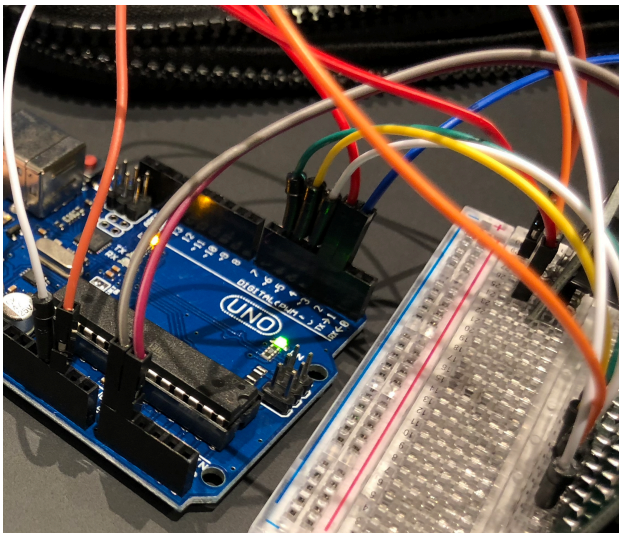


3.The five jumper wires from the matrix to the Uno are placed next (Wires to D4/D5/D6 are fixed neatly in parallel).

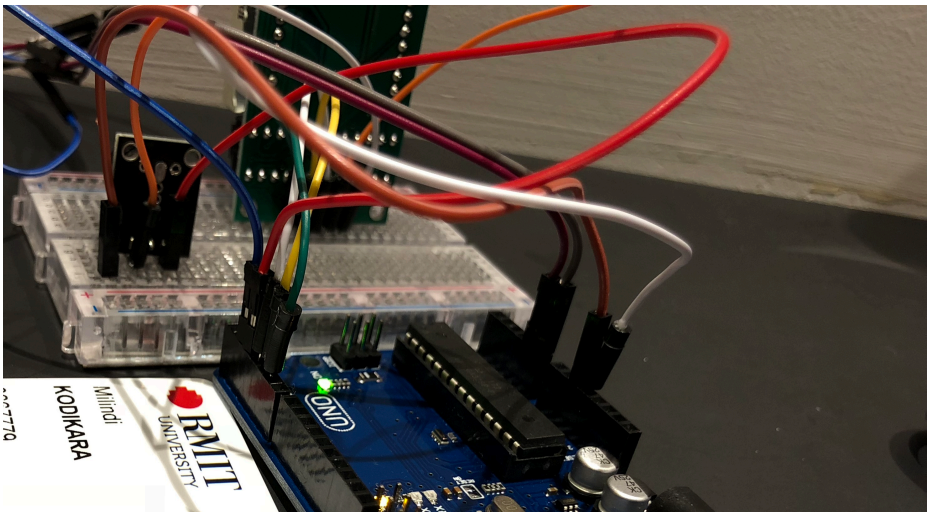




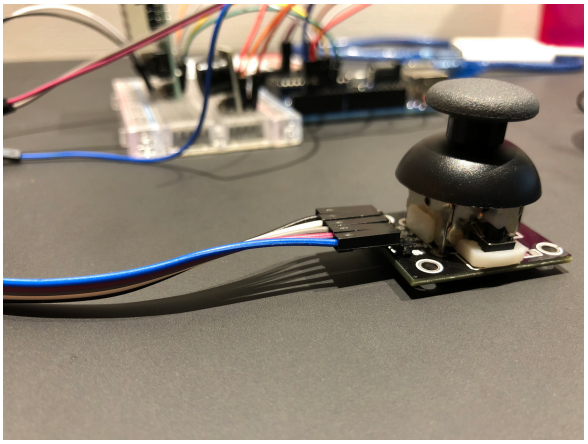
4. Fix the GND wire of the matrix to the - on the buzzer and then from there to the GND on the Uno.

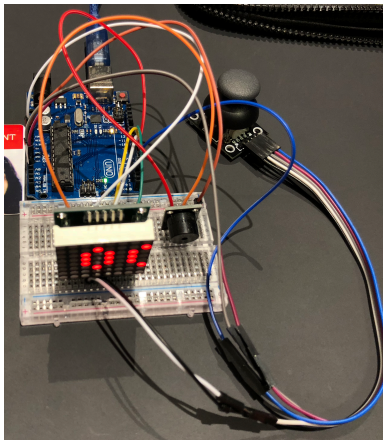


5.A jumper wire was fixed from the S on the buzzer to D3.

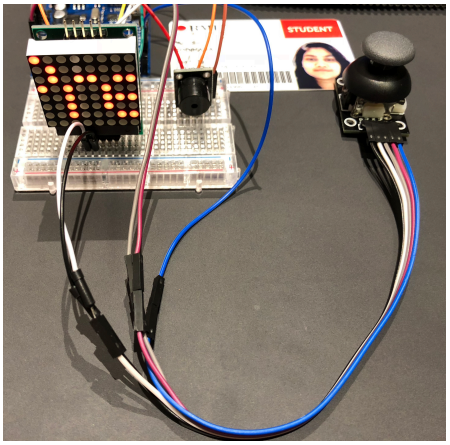


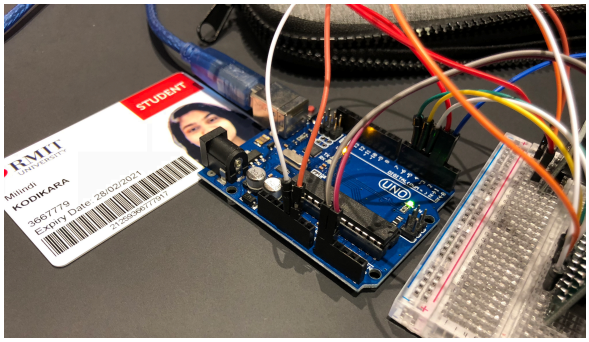
6.In order to move the joystick module easily, the flexibility of usage was increased by plugging in a group of five female - female jumper wires into the headers on the joystick module. Male - male jumper wires were used to extend from the female end of the afore mentioned jumper wires.

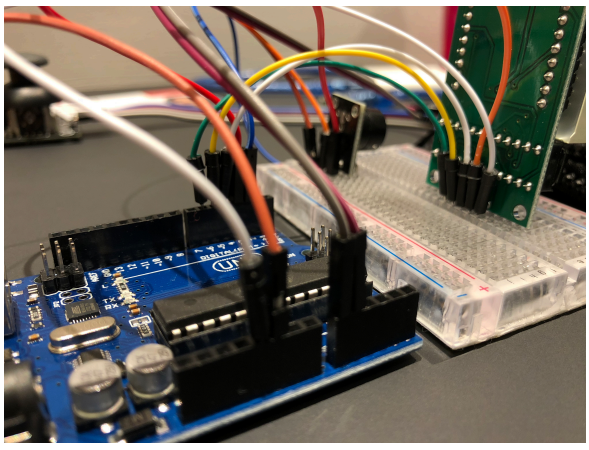




7.The joystick is then connected to D2,A0 and A1 of the Arduino board along with the GND and +5V connections.







Operation

Initially the player is greeted with a “Press to start” message on the display along with a melody. When the player clicks down on the joystick module the game starts. A tone is played so as to notify the player the start of the game. The snake is steered using the joystick module. During the game, a second tone is generated in time with the movement of the snake. The game is designed in a way that when the snake crashes into a wall or itself, the game ends. The player is notified of the end of the game using a third tone.The score of the player is displayed after the tone has been played along with a second melody. Finally, when the player clicks on the joystick module again, the game restarts and the welcome message is displayed on the matrix.

Modifications

I have modified the existing project in the following ways:
The code which is quite long, has been broken down into many smaller, more manageable parts in order to clearly understand. The speed of the game is set by the variable “tc” which I have modified to suit my purpose in the setup() and the loop() functions.

Furthermore, I have modified the welcome message as well as the display of the score by adjusting the array containing the displays to the matrix. The speed of the horizontal scrolling process of the words have been modified so that the player can read the greeting and the score easily.

As I have added the buzzer module, apart from the tone in rhythm with the movement of the snake during the game, I have added different melodies which are played along with the display; at the start of the game as well as at the end of the game. I have added a new library named “ themes.h” which contains the notes of the melodies played. Furthermore, I have added small tones of different time durations so as to signal the player the start of the game and the end of the game.

Code

The code used for this project along with the extra library, “themes.h” used for the melodies is depicted below. [The modifications are written in blue]

```
#define MAX7219DIN 4
#define MAX7219CS 5
#define MAX7219CLK 6
#define JOYX A1
#define JOYY A0
#define BUTTON 2
#define SOUND 3

#include "pitches.h"
#include "themes.h"


byte p[8]={0,0,0,0,0,0,0,0};      //bitmap to display
byte nums[10][3]={                //bitmap of numbers
  {28,34,28},
  {36,62,32},
  {50,42,36},
  {42,42,20},
  {28,18,62},
  {46,42,18},
  {28,42,18},
  {50,10,6},
  {54,42,54},
  {4,42,28}
};

byte marquee1[]={0,0,0,0,0,0,0,0,0,127,9,9,6,120,4,4,56,84,84,24,0,72,84,36,0,72,84,36,0,0, 0,  62, 68, 68, 56, 68, 68, 56, 0,  0,  72, 84, 36, 0,  62, 68, 68, 0,  116,  84, 120,  0,  120,  4,  4,  0,  62, 68, 68, 0,  0,  0,  0,  0,  0,  0,  0,  0,  0};
byte
marquee2[]={0,0,0,0,0,0,70,73,49,0,56,68,68,0,56,68,68,56,0,120,4,4,0,60,84,92,0,40,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

char sx[64];
char sy[64];                  //snake positions, zeroth is front of snake, first slen are valid
char dir=0;                   //1=up, 2=right, 3=down, 4=left
char slen=0;
int score=0;
char gamestate=0;             //0=idle, 1=play, 2 =game over
unsigned long tc=1400;          //time between game cycles (~speed/difficulty)
```

```
void setup() {
  MAX7219init();
  MAX7219brightness(1);
  pinMode(BUTTON,INPUT_PULLUP);
  pinMode(SOUND,OUTPUT); //Output of sound through the buzzer

}

void loop() {
        switch(gamestate){
                case 0: doidle(); break;
                case 1: dogame(); break;
                case 2: dogameover(); break;
                case 3: dogameend(); break;
 }
  delay(tc);
}

void doidle(){

        for (int thisNote = 0; thisNote < (sizeof(Pirates_note)/sizeof(int)); thisNote++) {
                int noteDuration = 1000 / Pirates_duration[thisNote];//convert duration to time delay
                tone(3, Pirates_note[thisNote], noteDuration);

                int pauseBetweenNotes = noteDuration * 1.05; //Here 1.05 is tempo, increase to play it slower
                delay(pauseBetweenNotes);
                noTone(3); //stop music on pin 3
        tc=200; //change the speed of the words
        static int n=0;
        MAX7219sendbm(marquee1+n);
        n++;

         if(n>sizeof(marquee1)-8){n=0;}
         if(digitalRead(BUTTON)==LOW){   //change modes,
                gamestate=1;
                startMelody();//Meldoy to signal the start of the game
                MAX7219sendbm(marquee1); //blank screen
                tc=200;                  //slow down
                slen=0;  //reset game
                while(digitalRead(BUTTON)==LOW){}// wait til released
                break;
                 }
         }
}
```

```
void dogame(){
        int i;          //loop variable
        char newx,newy;
        int a;
        if(slen==0){    //game is reset > initialise
                score=0;
                dir=1;
                sx[0]=3;
                sy[0]=3;
                sx[1]=3;
                sy[1]=4;
                sx[2]=3;
                sy[2]=5;
                slen=3;
        }
        newx=sx[0];
        newy=sy[0];   //new snake head position
        a=analogRead(JOYX);
        if(a<256){dir=2;}
        if(a>768){dir=4;}
        a=analogRead(JOYY);
        if(a<256){dir=1;}
        if(a>768){dir=3;}
        switch(dir){
                case 1: newy=newy-1;break;
                case 2: newx=newx+1;break;
                case 3: newy=newy+1;break;
                case 4: newx=newx-1;break;
        }
        if((newx<0)||(newx>7)||(newy<0)||(newy>7)){   //outside walls > game over
        gamestate=3;
        }
        for(i=0;i<slen;i++){
        if((newx==sx[i])&&(newy==sy[i])){gamestate=3;}  //collided with self > game over
        }
        for(i=63;i>0;i--){ //move old positions
        sx[i]=sx[i-1];
        sy[i]=sy[i-1];
        }
        sx[0]=newx;
        sy[0]=newy;
        for(i=0;i<8;i++){p[i]=0;}//clear display
        for(i=0;i<slen;i++){//draw snake
        p[sx[i]]=p[sx[i]]|(1<<(sy[i]));
        }
        MAX7219sendbm(p);
        score++;//increase score
        slen=(score+50)/25; //increase length
        tone(SOUND,440,20);  //make a sound for game rhythm

}
```

```
void dogameover(){
        tc=180;
        static int n=0;
        marquee2[30]=nums[(score/100)%10][0];
        marquee2[31]=nums[(score/100)%10][1];
        marquee2[32]=nums[(score/100)%10][2];
        marquee2[34]=nums[(score/10)%10][0];
        marquee2[35]=nums[(score/10)%10][1];
        marquee2[36]=nums[(score/10)%10][2];
        marquee2[38]=nums[(score/1)%10][0];
        marquee2[39]=nums[(score/1)%10][1];
        marquee2[40]=nums[(score/1)%10][2];

        for (int thisNote = 0; thisNote < (sizeof(CrazyFrog_note)/sizeof(int)); thisNote++) {

                int noteDuration = 1000 / CrazyFrog_duration[thisNote]; //convert duration to time delay
                tone(3, CrazyFrog_note[thisNote], noteDuration);
                int pauseBetweenNotes = noteDuration * 1.30;//Here 1.30 is tempo, decrease to play it faster
                delay(pauseBetweenNotes);
                noTone(3); //stop music on pin 3
                MAX7219sendbm(marquee2+n);
                n++;
                if(n>sizeof(marquee2)-8){n=0;}
                if(digitalRead(BUTTON)==LOW){   //change modes,
                        gamestate=0;
                        MAX7219sendbm(marquee1); //blank screen
                         endMelody(); //Note played to notify the restart of the game
                        while(digitalRead(BUTTON)==LOW){}// wait til released
                break;
                }
        }
}


void dogameend(){
        gamestate=2;
        endMelody(); //The sound made when the game is lost
}
```

```
void startMelody(){
            int melody[] = {
            NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
           };

           // note durations: 4 = quarter note, 8 = eighth note, etc.:
           int noteDurations[] = {
           4, 8, 8, 4, 4, 4, 4, 4
           };
           for (int thisNote = 0; thisNote < 8; thisNote++) {
                       int noteDuration = 1000 / noteDurations[thisNote];
                        tone(3, melody[thisNote], noteDuration);
                       int pauseBetweenNotes = noteDuration * 1.30;
                        delay(pauseBetweenNotes); // stop the tone playing:
                       noTone(8);
            }
}

void endMelody(){
            int melody[] = {
            NOTE_AS4, NOTE_A4,
            NOTE_GS4, NOTE_DS4, NOTE_B3,  NOTE_AS3, NOTE_A3, NOTE_GS3};

           int noteDurations[] = {
           12, 12, 6, 6, 18, 18, 18,
            6
           };
           for (int thisNote = 0; thisNote < 8; thisNote++) {
                       int noteDuration = 1000 / noteDurations[thisNote];
                       tone(3, melody[thisNote], noteDuration);
                       int pauseBetweenNotes = noteDuration * 1.30;
                       delay(pauseBetweenNotes);  // stop the tone playing:
                       noTone(8);
            }
}


void MAX7219shown(byte n){
           byte s[8];
           s[0]=nums[(n/10)%10][0];
           s[1]=nums[(n/10)%10][1];
           s[2]=nums[(n/10)%10][2];
           s[3]=0;
           s[4]=nums[n%10][0];
           s[5]=nums[n%10][1];
           s[6]=nums[n%10][2];
           s[7]=0;
           MAX7219sendbm(s);
           }

void MAX7219sendbm(byte p[]){
           for(int i=0;i<8;i++){
           MAX7219senddata(i+1,p[i]);
           }
```

```
}

void MAX7219brightness(byte b){  //0-15 is range high nybble is ignored
        MAX7219senddata(10,b);        //intensity
}

void MAX7219init(){
        pinMode(MAX7219DIN,OUTPUT);
        pinMode(MAX7219CS,OUTPUT);
        pinMode(MAX7219CLK,OUTPUT);
        digitalWrite(MAX7219CS,HIGH);   //CS off
        digitalWrite(MAX7219CLK,LOW);   //CLK low
        MAX7219senddata(15,0);        //test mode off
        MAX7219senddata(12,1);        //display on
        MAX7219senddata(9,0);         //no decode
        MAX7219senddata(11,7);        //scan all
        for(int i=1;i<9;i++){
                MAX7219senddata(i,0);       //blank all
         }
}


void MAX7219senddata(byte reg, byte data){
         digitalWrite(MAX7219CS,LOW);   //CS on
         for(int i=128;i>0;i=i>>1){
                  if(i&reg){
                           digitalWrite(MAX7219DIN,HIGH);
                  }else{
                           digitalWrite(MAX7219DIN,LOW);
                  }
        digitalWrite(MAX7219CLK,HIGH);
        digitalWrite(MAX7219CLK,LOW);   //CLK toggle
         }
        for(int i=128;i>0;i=i>>1){
         if(i&data){
                digitalWrite(MAX7219DIN,HIGH);
        }else{
                digitalWrite(MAX7219DIN,LOW);
        }
         digitalWrite(MAX7219CLK,HIGH);
         digitalWrite(MAX7219CLK,LOW);   //CLK toggle
        }
        digitalWrite(MAX7219CS,HIGH);   //CS off
}
```

```
int Pirates_note[] = {
NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4,
NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4,
NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_D4,
NOTE_A3, NOTE_C4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_E4, NOTE_F4, NOTE_F4,
NOTE_F4, NOTE_G4, NOTE_E4, NOTE_E4, NOTE_D4, NOTE_C4, NOTE_C4, NOTE_D4,
0, NOTE_A3, NOTE_C4, NOTE_B3, NOTE_D4, NOTE_B3, NOTE_E4, NOTE_F4,
NOTE_F4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4,
NOTE_D4, 0, 0, NOTE_A3, NOTE_C4, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_F4,
NOTE_G4, NOTE_G4, NOTE_G4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_A4, NOTE_G4,
NOTE_A4, NOTE_D4, 0, NOTE_D4, NOTE_E3, NOTE_F4, NOTE_F4, NOTE_G4, NOTE_A4,
NOTE_D4, 0, NOTE_D4, NOTE_F4, NOTE_E4, NOTE_E4, NOTE_F4, NOTE_D4
};
int Pirates_duration[] = {
4,8,4,8,4,8,8,8,8,4,8,4,8,4,8,8,8,8,4,8,4,8,
4,8,8,8,8,4,4,8,8,4,4,8,8,4,4,8,8,
8,4,8,8,8,4,4,8,8,4,4,8,8,4,4,8,4,
4,8,8,8,8,4,4,8,8,4,4,8,8,4,4,8,8,
8,4,8,8,8,4,4,4,8,4,8,8,8,4,4,8,8
};


int CrazyFrog_note[] = {
  NOTE_D4, 0, NOTE_F4, NOTE_D4, 0, NOTE_D4, NOTE_G4, NOTE_D4, NOTE_C4,
  NOTE_D4, 0, NOTE_A4, NOTE_D4, 0, NOTE_D4, NOTE_AS4, NOTE_A4, NOTE_F4,
  NOTE_D4, NOTE_A4, NOTE_D5, NOTE_D4, NOTE_C4, 0, NOTE_C4, NOTE_A3, NOTE_E4, NOTE_D4,
  0,NOTE_D4,NOTE_D4
};
int CrazyFrog_duration[] = {
  8, 8, 6, 16, 16, 16, 8, 8, 8,
  8, 8, 6, 16, 16, 16, 8, 8, 8,
  8, 8, 8, 16, 16, 16, 16, 8, 8, 2,
  8,4,4
};
```

References

Saddam, Snake Game on 8*8 Matrix using Arduino , CircuitDigest , viewed 10 May 2018,<https://circuitdigest.com/microcontroller-projects/arduino-snake-game-using-8x8-led-matrix>

Dilip Raja, 8*8 LED Matrix using Arduino, CircuitDigest, viewed 14 May 2018,<https://circuitdigest.com/microcontroller-projects/arduino-8x8-led-matrix>

Arduino 8*8 LED Matrix, Electronics Hub, viewed 15 May 2018,< https://www.electronicshub.org/arduino-led-matrix/ >

Nick Koumaris, Driving an 8*8 LED Matrix with MAX7219(or MAX7221) and Arduino Uno, Electronics Lab, viewed 16 May 2018,<http://www.electronics-lab.com/project/driving-8x8-64-led-matrix-max7219-max7221-arduino-uno/>

Arduino, Adeept, viewed 17 May 2018, <http://www.adeept.com>

Abrezengineer , Arduino Insight - Beginner LED's and Binary counter 16 BIT, Instructables, viewed 19 May 2018,<http://www.instructables.com/id/Arduino-Insight-LEDS/>

Arduino Uno 2018, Wikipedia:The Free Encyclopedia, Wikimedia Foundation Inc., viewed 20 May 2018, <https://en.wikipedia.org/wiki/Arduino_Uno>

B.Ashwinth Raj, Playing Melodies using Tone () Function, CircuitDigest, viewed 20 May 2018, <https://circuitdigest.com/microcontroller-projects/playing-melodies-on-piezo-buzzer-using-arduino-tone-function>

XC3900 Mini-project - Snake Game, JayCar Electronics, viewed 22 May 2018,<https://www.jaycar.com.au/arduino>

VFD Tube, n.d. photograph , viewed 25 May 2018 ,<http://2manyprojects.net/arduino-starter-kit/arduino-project-ideas/ >

Merry Light, n.d. photograph, viewed 25 May 2018, <https://www.aliexpress.com/item/P7-62-RB-P7-62-RB-Dot-Matrix-LED-Text-Display-16-128pixels-2pcs-lotDot-Matrix/469966632.html>