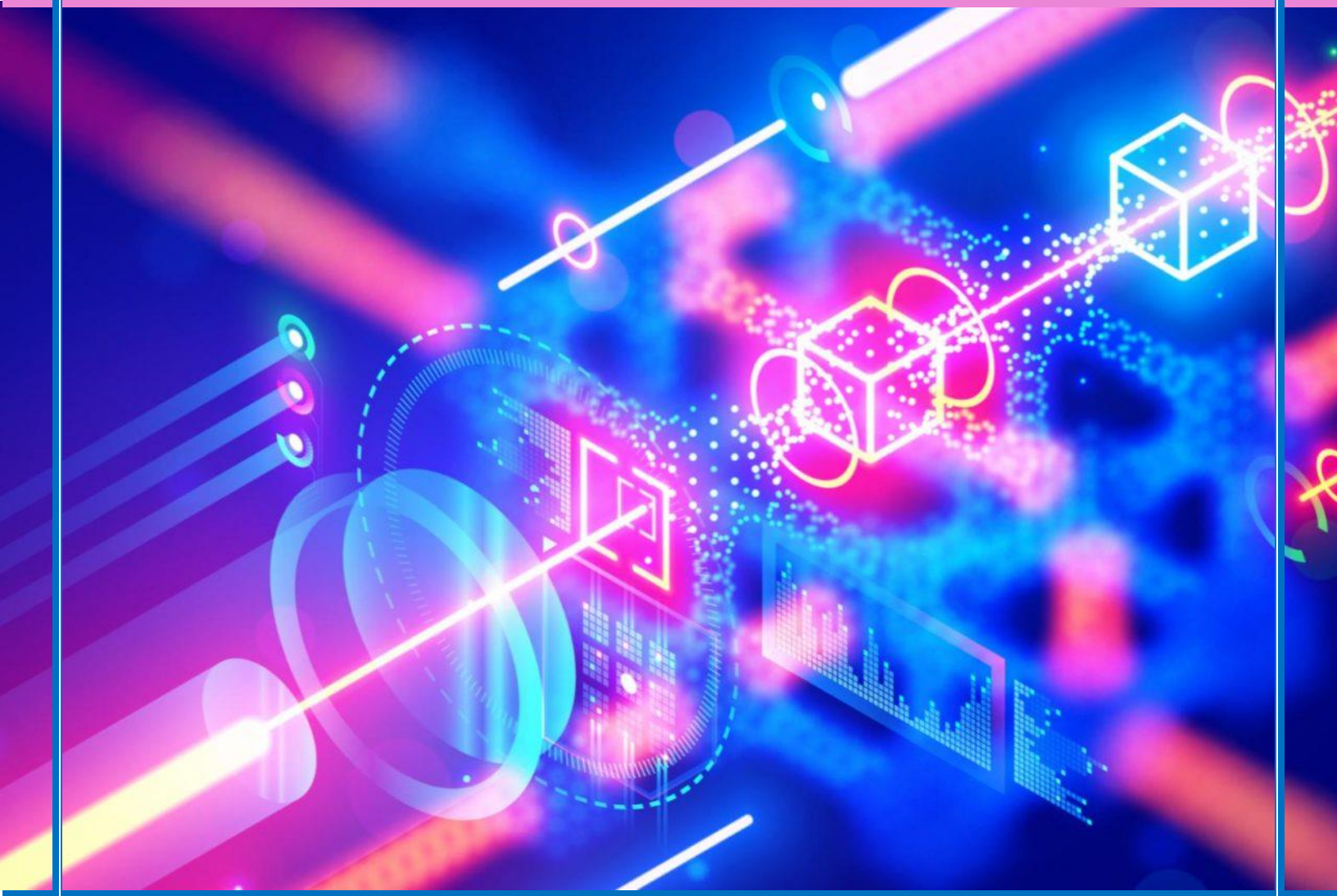
19102697 nourhane abdelalem| |19102749 bassant yasser

19103642 seham mahmoud || 19100683 yara adel hassan



Project Assignment
CS427 Embedded Systems

Supervised by: Dr. /Mohamed habrouk

Endless runner

Documentation report

Our describtion:

Project idea is dependant upon the usage of an arduio bored as a key pad for games usage

And dependant upon the usage of the lcd as a computer screen throught which our game will be displayed

Our game was built on limited resources to ensure that it will be in expensive

Was made for the sake of entertainment and appliying what we have learned through the course.

Our components:

	Resistor 330 Ohm 0.25W Resistor 1 Kohm 0.25W Resistor 10 Kohm 0.25W Resistor 100 Kohm 0.25W
	Liquid Crystal Display - LCD 16x2 (with Header Pins)
	Potentiometer 100k
	Potentiometer Cap
	USB Cable for Arduino Uno 1.5m (Shielded)
	Arduino Uno Board

	Jumper Wires for Breadboard and Arduino (65 pcs)
	Breadboard 830 points

Our sample code:

```
#include <LiquidCrystal.h>
#define RS 6
#define E 7
#define D4 2
#define D5 3
#define D6 4
#define D7 5
#define Button 8  // the input button
#define COIN_CHANCE 50
#define SPIKE_CHANCE 75
#define WINNING_SCORE 5
LiquidCrystal lcd(RS,E,D4,D5,D6,D7) ;
// the lcd has 1280 pixle it can display 16x2= 32 char the one char takes 5x8 = 40 pixels
byte blank[]{
```

```
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
};
byte hero_1[]{  // the part 1 of the walk cycle walking leg forwrd
  B00100,
  B01110,
  B01110,
  B00100,
  B01110,
  B00100,
  B00110,
  B01001,
};
byte hero_2[]{  // the part 2 of the walk cycle  walking leg back
  B00100,
  B01110,
  B01110,
  B00100,
  B01110,
  B00100,
  B01100,
  B10010,
};
byte spike_s[]{  // small spike
  B00000,
  B00000,
  B00000,
```

```
  B00000,
  B00000,
  B00000,
  B00100,
  B01110,
};
byte spike_m[]{  //midium spike
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00100,
  B01110,
  B11111,
};
byte spike_l[]{  //large spike
  B00000,
  B00000,
  B00000,
  B00100,
  B00100,
  B01110,
  B01110,
  B11111,
};
byte coin[]{
  B00000,
  B00100,
  B01010,
  B01010,
  B01010,
  B00100,
```

```
  B00000,
  B00000,
};
byte flag[]{
  B11111,
  B11010,
  B10101,
  B11010,
  B11111,
  B10000,
  B10000,
  B10000,
};
// the row states in the begining of the game and also is used for reseting the game
int init_row_0 [] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} ;
int init_row_1 [] = {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0} ; // 0 means empty space and 1 is the hero on the lcd 16 characters

// the rows that maintains the current state of the game
int row_0 [] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} ;  //conatain the
int row_1 [] = {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0} ;

int score = 0 ;
bool game_active = true ;
bool win = false  ;
bool just_after = false ;

int jump_counter = 0 ;   // keeps track of time
int internal_counter = 0 ;

void row_0_update(){
  int hero_idx = -1  ;
  for(int i=0 ; i< 16 ; i++){ //search for the hero in the row if not found the set the index to -1
```

```
  if (row_0[i] == 1 || row_0[i] == 2 ){
    hero_idx = i ;
   }
 }
 if(internal_counter == 0 && hero_idx !=-1 ){ // swap the walk cycle of the her0
  if(row_0[hero_idx] == 1){
     row_0[hero_idx] = 2 ;
  }
  else{
     row_0[hero_idx] = 1 ;
  }
 }
 if(jump_counter == 0 && hero_idx !=-1 ){ // if the hero is at the row zero and we need to drop them
into the row below
    row_1[hero_idx] = row_0[hero_idx];
    row_0[hero_idx] = 0 ;
    hero_idx = -1 ;
 }
 for(int i=0 ; i< 16 ; i++){ // check collision between the hero and the coin
  if (row_0[i] == 6 ){
    if((i-1) == hero_idx && hero_idx !=-1){ // check the hero position
      score++ ;
      row_0[i] = 0 ;  // make the coin disappear
    }
  }
  //move the objects closer to the row
 if(row_0[i] > 2 && internal_counter ==  0){
    row_0[i-1] = row_0[i] ;
    row_0[i] =0  ;
 }
 if(internal_counter == 0 && random(COIN_CHANCE) == 1 && !(row_0[13] == 6 && row_0[14]== 6
&& row_0[15] == 6 )){
  row_0[15] = 6 ;
 }
```

```
  }


}
void row_1_update(){
  int hero_idx = -1  ;
  for(int i=0 ; i< 16 ; i++){ //search for the hero in the row if not found the set the index to -1
    if (row_1[i] == 1 || row_1[i] == 2 ){
      hero_idx = i ; }
  }
  if(internal_counter == 0 && hero_idx !=-1 ){ // swap the walk cycle of the her0
    if(row_1[hero_idx] == 1){
       row_1[hero_idx] = 2 ;
    }
    else{
       row_1[hero_idx] = 1 ;
    }
  }
  if(jump_counter > 0 && hero_idx !=-1 ){ // if the hero is at the row zero and we need to drop them
into the row below
     row_0[hero_idx] = row_1[hero_idx];
     row_1[hero_idx] = 0 ;
     hero_idx = -1 ;
  }


  for(int i=0 ; i< 16 ; i++){
    //collision with the spike
    if(row_1[i] == 3 || row_1[i]==4 || row_1[i]==5 ){
       if((i-1) == hero_idx && hero_idx > 0 ){
         game_active = false ;
         win = false ;
         just_after = true ;
         row_0[i] = 0 ;
```

```
      }
    }
    //move the objects closer to the row


  if(row_1[i] == 7){
     if((i-1) == hero_idx && hero_idx > 0 ){
       game_active = false  ;
       win = true ;
       just_after = true ;
       row_0[i] = 0  ;
     }
  }


   if(row_1[i] > 2 && internal_counter ==  0){
      row_1[i-1] = row_1[i] ;
      row_1[i] = 0 ;
  }



  if(score < WINNING_SCORE){
     if(internal_counter == 0 && random(SPIKE_CHANCE) == 1 && !(row_1[14]==4 &&
row_1[15]==4)){
       row_1[15] = 4 ;
     }
  }
  else{
     row_1[15] = 7 ;
  }


  }
}
```

```
void ctrl_update(){
 int Button_val = digitalRead(Button) ;
 if(Button_val== LOW && jump_counter == 0 ){ // it means that the button has just been realsed
   if(game_active){
     jump_counter = 10 ;  // the player will be on the air for ten loops
   }
   else{ // pushing the button means start the game
     game_active = true ;
     win = false ;
     just_after = false ;
     score = 0 ;

   for(int i=0 ; i<16 ; i++){
    if(i== 2 ){
      row_0[i] = 0 ;
      row_1[i] = 1 ;

    }
    else {
      row_0[i] = 0 ;
      row_1[i] = 0  ;
    }
   }
 }
 }
}

void ui_update(){
 lcd.home() ;
 for(int i=0 ; i<14; i++ ){ // this is the first raw data
   lcd.print((char)row_0[i]) ;
 }
 lcd.setCursor(0, 1) ;
```

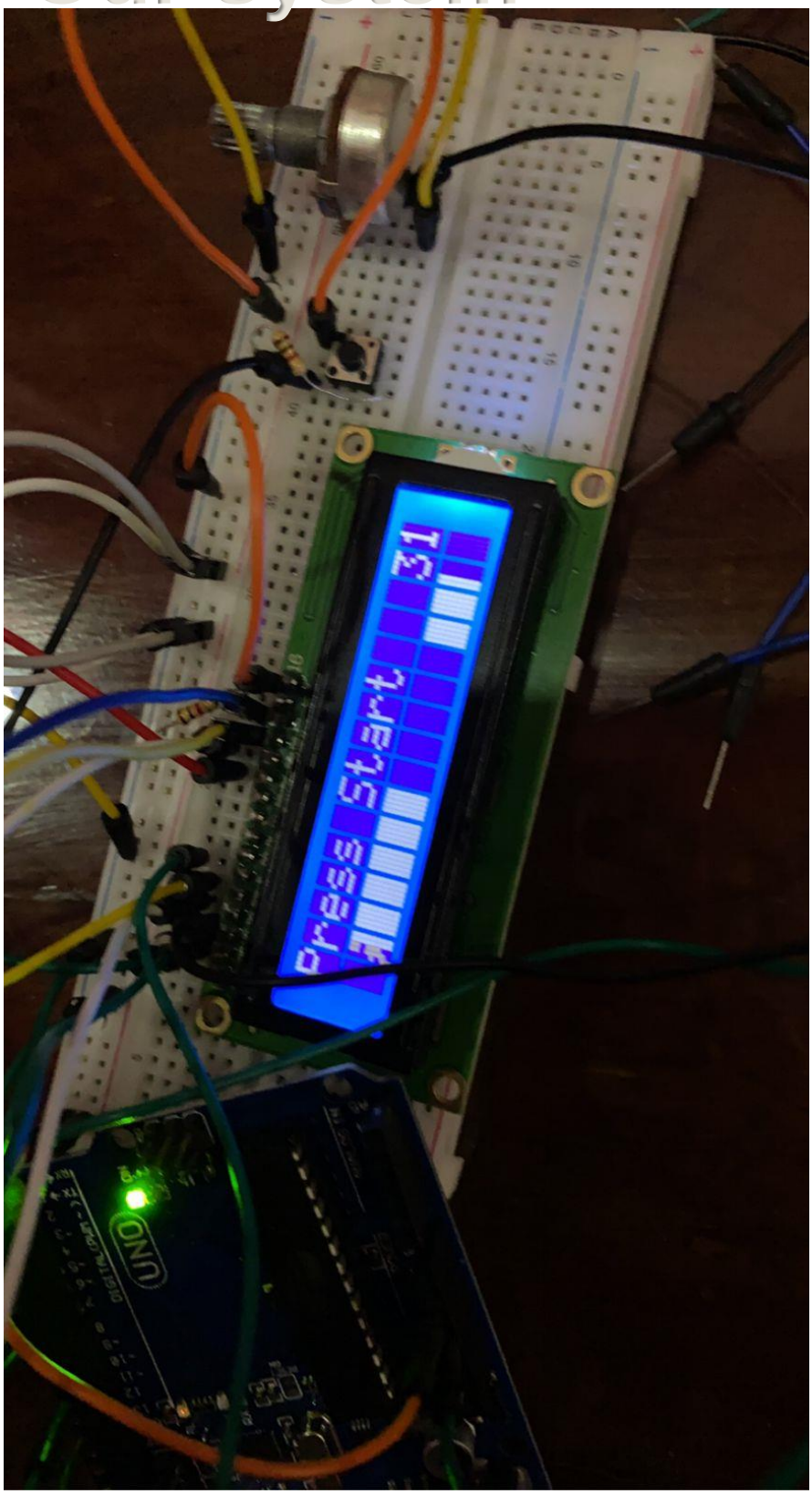
```
 for(int i=0;i<16;i++){
   lcd.print((char)row_1[i]) ;
 }
 lcd.setCursor(14,0) ;
 if(score < 10){ //in case the score is less than two digits we print zero ahead of it
   lcd.print("0");
 }
 lcd.print(score) ;
}


void setup() {
  Serial.begin(9600) ;
  pinMode(Button,INPUT_PULLUP) ;
  lcd.begin(16,2) ;
  randomSeed(analogRead(A0)); //make random shapes of the spikes every time we start the game
but we use seed to be the volt on the A0 pin which is not conncected so we will get random
voltages every time
  lcd.createChar(0,blank) ;
  lcd.createChar(1,hero_1) ;
  lcd.createChar(2,hero_2) ;
  lcd.createChar(3,spike_s) ;
  lcd.createChar(4,spike_m) ;
  lcd.createChar(5,spike_l) ;
  lcd.createChar(6,coin) ;
  lcd.createChar(7,flag) ;
  lcd.clear() ;
  lcd.home() ; //put the cursor at the begining of the screen
  //bgining of the game
  lcd.print("-ENDLESS RUNNER-") ;
  lcd.setCursor(0,1) ;  // set the cursor at the begining of the second row
  lcd.print("Try to score 5") ;
  delay(2500) ;
  lcd.clear() ;
```

```
}

void loop() {
  if(game_active){ // game active is initially true
     if(jump_counter > 0 ){
       jump_counter -=1 ;
     }
     if (score > 99 ){
       score = 0 ;
     }
     if(internal_counter < 5 ){
       internal_counter++ ;
     }
     else{
       internal_counter =  0 ;
     }
     ctrl_update() ;
     row_0_update() ;
     row_1_update() ;
     ui_update() ;
  }
  else{
    if (just_after){
     delay(1000) ;
     lcd.clear() ;
     lcd.home() ;
    if(win){
     lcd.print("YOU WON ");
     lcd.setCursor(0,1)  ;
     lcd.print("Press to play ") ;
    }
    else{
     lcd.print("Game over") ;
```

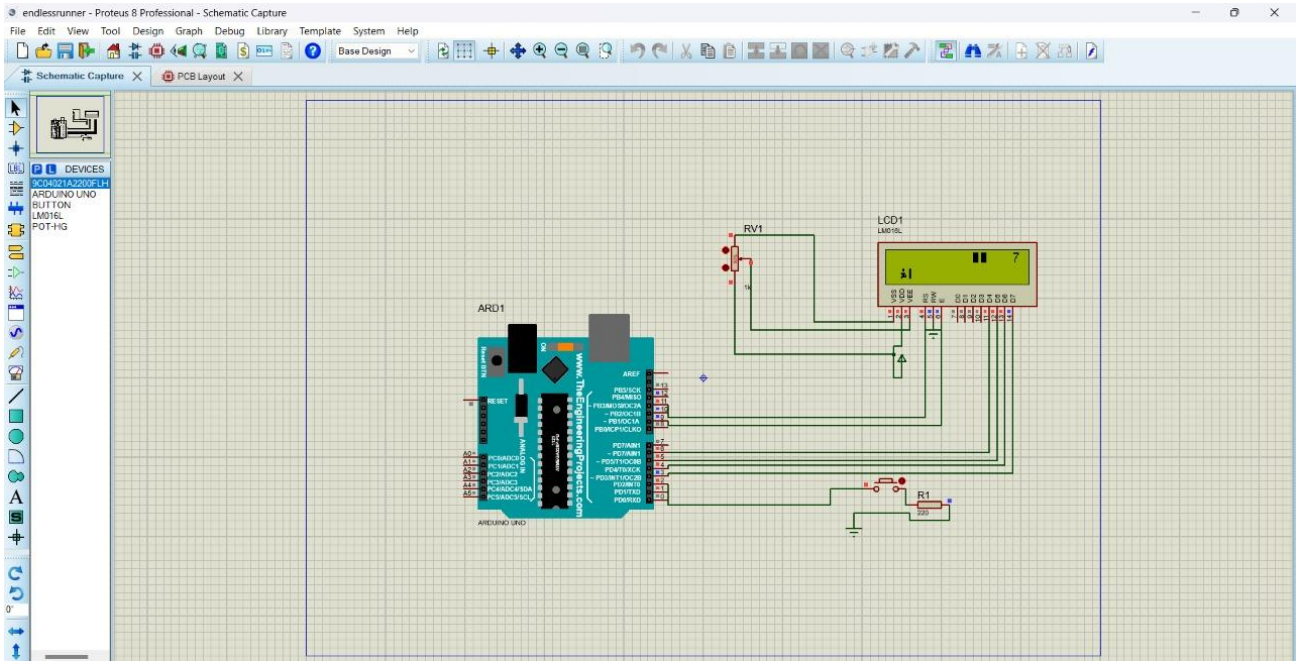
```
    lcd.setCursor(0,1) ;
    lcd.print("press to play ") ;
   }
    just_after = false ;
  }
  ctrl_update() ;  // take the input from the button

 }
 delay(100) ;
}}
```

Our poster:

Our system



Our simulation



Our testing plan

1- **Brain stormed the concept**
2- **Ideation and research**
3- **Writing the code**
4- **Cost analysis to choose the right component resources**
5- **Purchasing the components**
6- **Settling the bored**
7- **Uploading the code on the Arduino**

8- **Verifying the code “it runned succesfully”**
9- **Displaying the player and the obstacles**
10- **Testing the win and lose states**
11- **Setting a marketing plan for the product**

Our marketing plan

Since 2010 people intended to purchase more laptops than pcs as they where catchy but after 5 – 7 years people went back to their roots . statistics said pcs made a very good come back inn the market . which made software companies starts to re develop pcs to be more efficient than before

Facing trend strategies and according to marketing statistics games are less purchased nowadays due to high pricing so we managed to develop[most common games in a low price ith limited resources .