



---

## Implementarea jocului Pacman

---

Realizat de  
Şimonca Giorgiana-Florentina  
Grupa 30238

Indrumator  
Muresan Mircea Paul

Facultatea de Automatică și Calculatoare  
Departamentul de Calculatoare și Tehnologia Informației  
Materia : Proiectare cu Micro-PRocesoare  
07.01.2021

## Cuprins

<b>1 Rezumat</b>	<b>1</b>
<b>2 Introducere</b>	<b>2</b>
<b>3 Implementarea Circuitului Fizic</b>	<b>4</b>
<b>4 Implementare software</b>	<b>5</b>
4.1 Implementarea Codului . . . . .	7
<b>5 Testarea</b>	<b>12</b>
<b>6 Concluzii</b>	<b>13</b>
<b>7 Anexa A</b>	<b>14</b>

## **1 Rezumat**

Proiectul s-a concentrat pe proiectarea jocului Pacman folosind mediul de dezvoltare Arduino alaturi de o placa de dezvoltare Arduino. Aceasta implementare a constat din analiza componentelor principale ce constituie jocul apoi construirea unei strategii de abordare a problemei. Alaturi de placa de dezvoltare Arduino s-au folosit si difeite periferice care s-au conectat la placa pentru a putea face legatura cu exteriorul si pentru ca utilizatorul sau jucatorul sa poata interactiona cu jocul. Rezultatul fiind un joc care contine 3 nivale cu complexitatea diferita care se aproape cat mai mult de jocul original Pacman.

## 2 Introducere

Acest proiect constă în implementarea jocului Pacman folosind cunoștințele acumulate pe parcursul laboratoarelor din cadrul materiei "Proiectare cu Micro-Procesoare". Jocul pacman simulează un labirint prin care trebuie să treaca caracterul principal astfel încât să adune toate bucatile de mancare și să se ferească de fantome astfel încât să nu piarda jocul, odată cu adunarea tuturor bucătilor de mancare jocul este castigat.



FIGURE 1 – Pac-Man Game

In cele ce urmează vom prezenta o scurtă istorie a jocului. Initial jocul a fost conceput și lansat de către firma Namaco, firma japoneză, în anul 1980. Construcția jocului a început în anul 1979 cu o echipă de 9 oameni. După lansare jocul s-a bucurat de o popularitate mare în randul tinerilor, iar azi este considerat unul dintre cele mai bune jocuri video, producând un profit de peste 14 milioane de dolari anual.

Pentru implementarea acestui joc se va utiliza o placă de dezvoltare Arduino UNO R3, aceasta având o componentă hardware care constă dintr-o placă cu circuite programabile fizice care îl poate adăuga senzori și diferite periferice apoi și o componentă software care este reprezentată de programul pe care îl rulăm pe computerul nostru. Senzorii adăugati au rolul de a face legătura cu exteriorul și de a putea să interacționeze utilizatorul cu jocul creat pentru a putea manevra caracterul nostru principal : Pacman-ul.

Un Arduino este o placă de dezvoltare microcontroler open-source. Puteți folosi Arduino pentru a citi senzori și a controla lucruri precum motoare și lumini. Acest lucru vă permite să încărcați programe în această placă, cu care apoi se poate interacționa cu lucrurile din lumea reală. Cu aceasta, puteți face dispozitive care răspund și reacționează la lumea în ansamblu. Cea mai comună versiune a Arduino este Arduino Uno. Aceste forum este ceea ce vorbesc majoritatea oamenilor când se referă la un Arduino. În pasul următor, există o descriere mai completă a caracteristicilor sale.

Unii oameni se gândesc la întreaga placă Arduino ca un microcontroler, dar acest lucru este inexact. Placa Arduino este de fapt o placă de circuite special proiectată pentru programare și experimentare cu microcontrolere Atmel.

Lucrul plăcut pentru placa Arduino este că ea este relativ ieftină, se conectează direct în portul USB al unui computer și este ușor de configurat și folosit (în comparație cu alte plăci de dezvoltare).

Unele dintre caracteristicile cheie ale Arduino Uno includ : Tip open-source. Avantajul faptului că este open-source este că are o comunitate largă de utilizatori și de depanare. Acest lucru face mai ușor să găsiți pe cineva care să vă ajute să depanați proiectele. O interfață ușoară USB. Cipul de pe placă se conectează direct în portul USB și se înregistrează pe computer ca un port serial virtual. Acest lucru vă permite să interfațați cu el ca prin intermediul unui dispozitiv serial. Beneficiul acestei configurații constă în faptul că comunicarea serială este un protocol extrem de ușor (și testat în timp), iar conexiunea USB la calculatoarele moderne este într-adevăr convenabilă. Comandă foarte convenabilă a alimentării și reglarea tensiunii încorporată. Puteți conecta o sursă externă de alimentare de până la 12V și va regla atât la 5V cât și la 3,3V. Poate fi alimentată direct de pe un port USB fără nici o putere externă. Un creier microcontroller ușor de găsit și ieftin. Cipul ATmega328 se vinde pentru aproximativ 2,88 USD pe Digikey. Are nenumărate caracteristici hardware cum ar

fi cronometre, pini PWM, întreruperi externe și interne și mai multe moduri de sleep. Consultați fișa tehnică oficială pentru mai multe detalii. Un ceas de 16 MHz. Acest lucru nu-l face cel mai rapid microcontroller din jur, dar suficient de rapid pentru majoritatea aplicațiilor. 32 KB de memorie flash pentru stocarea codului. 13 pini digitali și 6 pini analogi. Acești pini vă permit să conectați hardware extern la Arduino. Acești pini sunt cheia pentru extinderea capacitatii de calcul a Arduino în lumea reală. Pur și simplu conectați dispozitivele și senzorii în prizele care corespund fiecărui pin și sunteți gata de lucru. Un conector ICSP pentru ocolirea portului USB și conectarea directă a dispozitivului Arduino ca dispozitiv serial. Acest port este necesar pentru a reîncărca cipul în cazul în care acesta nu mai poate vorbi cu computerul. Un LED pe placă atașat la pinul digital 13 pentru o depanare rapidă și ușoară a codului. și nu în ultimul rând, un buton pentru resetarea programului pe cip.

### 3 Implementarea Circuitului Fizic

Pentru a putea realiza un joc cat mai aproape de realitate am folosit urmatoarele componente pentru a implementa jocul Pacman : o placuta Arduino Uno R3, un LCD cu 20 de coloane si 4 randuri cu un modul I2C, 4 butoane, fire, breadboard si un buzzer activ.

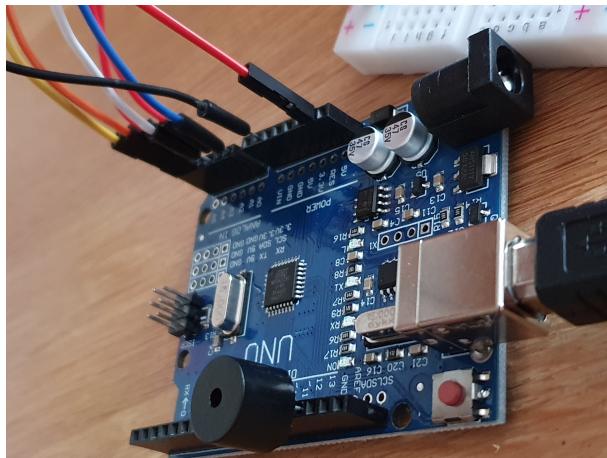


FIGURE 2 – RElizare circuit p1

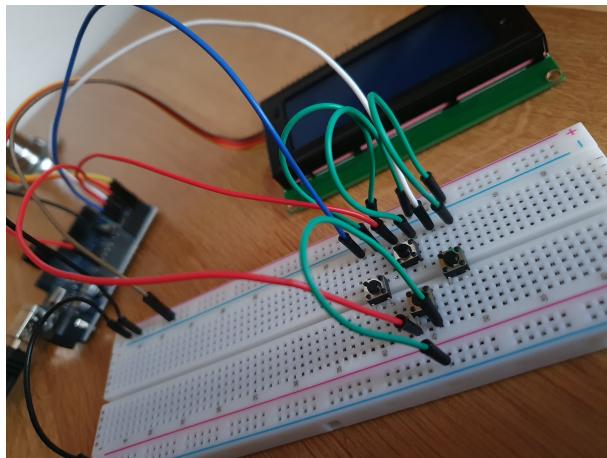


FIGURE 3 – Realizare circuit p2

In imaginile de mai sus se poate vedea implementarea fizica a componentelor si legarea lor la placa de dezvoltare Arduino Uno R3.

## 4 Implementare software

Pentru implementarea codului ce urmeaza sa controleze componentele hardware am folosit IDE-ul Arduino 1.0.6. Principala parte a acestei implementari consta in identificarea partilor principale ale jocului.

In primul rand odata cu intrarea in joc avem un mesaj specific cu un pacman care se misca. De asemenea, si la iesire avem un mesaj specific si intre cele 3 nivele avem un mesaj care anunta ce nivel urmeaza. Alte mesaje intrmediare ar fi mesajele pentru castigul nivelului si pentru pierderea unui nivel.



FIGURE 4 – Pac-Man Game



FIGURE 5 – Pac-Man Game

Pentru constructia celor 3 nivele am mers pe o crestere graduala a complexitatii. Astfel ca in continuare se pot vedea cele trei nivale. Pentru primul nivel am ales o complexitate foarte usoar ce consta doar din controlarea pacmanului de a ajunge la celalalt capat, nivelul fiind catigat cand toate bucatile de mancare au fost colectate.

Pentru cel de-al doilea nivel am adaugat si doua obstacole pentru a creste complexitatea. Si aici nivelul este castigat cand jucatorul colecteaza bucatile de mancare ferindu-se de cele 2 fantome.



FIGURE 6 – Pac-Man Game



FIGURE 7 – Pac-Man Game

Pentru cel de-al treilea nivel am crescut complexitatea deoarce numarul obstacolelor este mai mare, iar jucatorul trebuie sa se fereasca de fantome si sa colecteze toate bucatile de mancare.



FIGURE 8 – Pac-Man Game

Pentru constructia celor 3 nivele am fost limitata de numarul de randuri si coloane ale lcd-ului si nu am reusit sa fac niste frame-uri asemanatoare cu ale jocului original, dar consider ca este o reinterpretare destul de buna a jocului in care am adugat o nota personala. De asemenea, o alta limitare a fost faptul ca nu puteam sa fac si fantomele sa se miste in timp ce pacman-ul este controlat deoarce ese introduceau intarzieri sia vema nevoie de un thread special pentru fiecare fantoma care se le controleaza in paralelu cu controlul pacmanului.

#### 4.1 Implementarea Codului

Pentru partea de implementare a codului s-au folosit im majoritatea cazului functii care sunt specializate pentru executarea unui anumit task. Am ales aceasta varianta de a implementa codul cat mai structurat pentru a putea fi mai usor de depanat si de imblunatat de a fi mai usor sa adaugam functionalitati noi acolo unde se considera.

Pentru inceput se pot vedea toate variabilele de control globale care vor fi controlate de mai multe functii si caracterele speciale care vor fi salvate in memoria interna a LCD-ului. Aceste caractere se refera la pacman si la diferitele ziduri care ne ajuta sa construim diferite framework-uri.

```
byte pacman1[8] = {
    B00000,
    B00000,
    B01110,
    B11011,
    B11100,
    B11111,
    B01110,
};
```

De asemenea avem variabilele care sunt responsabile de pozitia Pacman-ului :

```
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
```

Majoritatea variabilelor au un nume sugestiv astfel ca este destul de usor de intuit pentru care parte a programului se folosesc si ce functionalitate controleaza.

Functiile de tipul de mai jos sunt functii care traseaza frame-ul jocului, avand 3 frameuri diferite toate implementate dupa felii de tipar.

```
void frame1() {
    for (int i = 0; i <= 18; i = i + 1) {
        lcd.setCursor(i, 0);
        lcd.write(byte(3));
    }
    for (int i = 0; i <= 18; i = i + 1) {
        lcd.setCursor(i, 3);
        lcd.write(byte(3));
    }

    lcd.setCursor(0, 1);
    lcd.write(byte(3));
    lcd.setCursor(0, 2);
    lcd.write(byte(3));

    for (int i = 1; i <= 19; i = i + 1) {
        lcd.setCursor(i, 1);
        lcd.write(byte(5));
        lcd.setCursor(i, 2);
        lcd.write(byte(4));
    }
}
```

Functiile pentru leve-uri sunt in continuarea acestor functii si au rolul de a lua decizii in functie de frame si de a transmite informatii la functiile care controleaza Pacmanul. Functiile care se ocupă de verificarea ca Pacman-ul sa respecte frame-ul sunt urmatoarele :

```
int verifyFrame1(int x, int y) {
    for (int i = 6; i <= 19; i = i + 1) {
        if (x == i && y == 1) {
            return 0;
        }
    }
    for (int i = 6; i <= 19; i = i + 1) {
        if (x == i && y == 2) {
            return 0;
        }
    }
    return 1;
}
```

Functiile de aspect sunt cele care se ocupă de generarea sunetului si de afisare mesajelor intermediare.

```
void you_won() {
    lcd.setCursor(0, 1);
    lcd.print("          YOU");
```

```

lcd.setCursor(0, 2);
lcd.print("          WON");
win_song();
}

////////// Cantecele pentru Buzzer

void win_song() {
    tone(buzzer, 420, 200);
    delay(500);
    tone(buzzer, 440, 200);
    delay(500);
    tone(buzzer, 490, 270);
    delay(300);
    tone(buzzer, 520, 200);
    delay(300);
}

```

Functiile care se ocupa de luarea deciziilor in ceea ce priveste mutarea pacmanului si afisarea pe lcd sunt urmatoarele.

```

void move_Pac(int state) {
    switch (state) {
        case rightBtn:
            v = verify(xPac + 1, yPac);
            if (v == 1) {
                xPac = xPac + 1;
                move_rigth(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case upBtn:
            v = verify(xPac, yPac - 1);
            if (v == 1) {
                yPac = yPac - 1;
                move_up(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case downBtn:
            v = verify(xPac, yPac + 1);
            if (v == 1) {
                yPac = yPac + 1;
                move_down(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case leftBtn:
            v = verify(xPac - 1, yPac);

```

```

if (v == 1) {
    xPac = xPac - 1;
    move_left(xPac, yPac, pacState);
    pacState = ~pacState;
    delay(300);
}
break;
default:
    // statements
    break;
}

////////// functiile pentru mutarea Pacman-ului in SUS,JOS,STANGA,DREAPTA

void move_down(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x, y - 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x, y - 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(0));
    }
}

void move_up(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x, y + 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x, y + 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(0));
    }
}

void move_left(int x, int y, int s) {

```

```

if (s == 0) {
    delay(200);
    lcd.setCursor(x + 1, y);
    lcd.print(" ");
    delay(200);
    lcd.setCursor(x, y);
    lcd.write(byte(1));
} else {
    delay(200);
    lcd.setCursor(x + 1, y);
    lcd.print(" ");
    delay(200);
    lcd.setCursor(x, y);
    lcd.write(byte(0));
}
}

void move_right(int x, int y, int s) {
if (s == 0) {
    delay(200);
    lcd.setCursor(x - 1, y);
    lcd.print(" ");
    delay(200);
    lcd.setCursor(x, y);
    lcd.write(byte(1));
} else {
    delay(200);
    lcd.setCursor(x - 1, y);
    lcd.print(" ");
    delay(200);
    lcd.setCursor(x, y);
    lcd.write(byte(0));
}
}

```

Restul functiilor vor fi acoperite in video.

## 5 Testarea

Înainte de a începe să testați ceva cu Arduino, trebuie să descărcați și să instalați IDE Arduino (mediu de dezvoltare integrat). Din acest punct vom vorbi despre IDE Arduino ca programator Arduino. Programatorul Arduino se bazează pe Processing IDE și utilizează o variantă a limbajelor de programare C și C++.

Conectați Arduino la portul USB al computerului.

Rețineți că, deși dispozitivul Arduino se conectează la computer, acesta nu este un dispozitiv USB adevărat. Placa are un cip special care îi permite să apară pe computer ca un port serial virtual când este conectat la un port USB. Aceasta este motivul pentru care este important să conectați placa. Atunci când placa nu este conectată, portul serial virtual pe care funcționează Arduino nu va fi prezent (deoarece toate informațiile despre el trăiesc pe placa Arduino).

De asemenea, este bine să știți că fiecare placă Arduino are o adresă unică a portului serial virtual. Acest lucru înseamnă că, de fiecare dată când conectați o placă Arduino diferită în computer, va trebui să reconfigurați portul serial care este utilizat.

Dupa se poate uploada codul din anexa A și odata cu relizarea circuitului se poate încărca proiectul pe placă și este gata de testare.

## **6 Concluzii**

In acest raport al proiectului a fost prezentata implementarea jocului Pacman folosind o placă de dezvoltare Arduino Uno R2 împreună cu alte periferice și folosind IDE-ul Arduino 1.0.6. Implementarea jocului încearcă o abordare cât mai aproape de realitate, dar din cauza posibilităților restrânse de programare am ajuns la o variantă mai simplistică, dar decentă în ceea ce privește jocul Pacman.

## 7 Anexa A

Aceasta anexa contine codul proiectului PACMAN.

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);
////////////////Caracterele pentru pacman, fantome, zid si mancare
byte pacman1[8] = {
    B00000,
    B00000,
    B01110,
    B11011,
    B11100,
    B11111,
    B01110,
};

byte pacman2[8] = {
    B00000,
    B00000,
    B01110,
    B11011,
    B11111,
    B11111,
    B01110,
};

byte ghost [8] = {
    B00000,
    B00000,
    B01110,
    B10101,
    B11111,
    B11111,
    B10101,
};

byte food [8] = {
    B00000,
    B00000,
    B00000,
    B00110,
    B00110,
    B00000,
    B00000,
};

byte zid1 [8] = {
    B11111,
```

```

B11111 ,
B00000 ,
B00000 ,
B00000 ,
B00000 ,
B00000 ,
B00000 ,
};

byte zid2[8] = {
    B00000 ,
    B00000 ,
    B00000 ,
    B00000 ,
    B00000 ,
    B00000 ,
    B11111 ,
    B11111 ,
};

byte zid3[8] = {
    B00011 ,
    B00011 ,
};

////varialbile pentru cele 4 butoane
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int buzzer = 11;

//nivelul curent
int nrLevel = 0;

//butonale
#define rightBtn A2
#define upBtn A1
#define downBtn A3
#define leftBtn A0

int xPac = 19; // pozitia initiala pe linie
int yPac = 0; // pozitia initiala pe coloana

void setup() {
    pinMode(buzzer , OUTPUT);
    pinMode(A0, INPUT_PULLUP);
}

```

```

pinMode(A1, INPUT_PULLUP);
pinMode(A2, INPUT_PULLUP);
pinMode(A3, INPUT_PULLUP);
lcd.begin();
lcd.createChar(0, pacman1);
lcd.createChar(1, pacman2);
lcd.createChar(2, ghost );
lcd.createChar(3, food );
lcd.createChar(4, zid1 );
lcd.createChar(5, zid2 );
lcd.createChar(6, zid3 );
program();
}

void loop() {}

//////////programul principal de unde gestioneaza functiile pentru nivele si mesajele de inc
void program() {
    lcd.clear();
    for (int i = 0; i < 2; i = i + 1) {
        start();
    }
    delay(500);
    lcd.clear();
    level1();
    delay(600);
    level2();
    delay(600);
    level3();
    delay(600);
    endGame();
    lcd.clear();
}
}

//////////construirea framurilor pentru cele 3 niveli
void frame1() {
    for (int i = 0; i <= 18; i = i + 1) {
        lcd.setCursor(i, 0);
        lcd.write(byte(3));
    }
    for (int i = 0; i <= 18; i = i + 1) {
        lcd.setCursor(i, 3);
        lcd.write(byte(3));
    }

    lcd.setCursor(0, 1);
    lcd.write(byte(3));
    lcd.setCursor(0, 2);
    lcd.write(byte(3));

    for (int i = 1; i <= 19; i = i + 1) {
        lcd.setCursor(i, 1);
    }
}

```

```

        lcd.write(byte(5));
        lcd.setCursor(i, 2);
        lcd.write(byte(4));
    }
}
void frame2() {
    for (int i = 0; i <= 9; i = i + 1) {
        lcd.setCursor(i, 1);
        lcd.write(byte(5));
        lcd.setCursor(i, 2);
        lcd.write(byte(4));
    }
    for (int i = 12; i <= 19; i = i + 1) {
        lcd.setCursor(i, 1);
        lcd.write(byte(5));
        lcd.setCursor(i, 2);
        lcd.write(byte(4));
    }
    lcd.setCursor(11, 1);
    lcd.write(byte(2));
    lcd.setCursor(11, 2);
    lcd.write(byte(2));

    for (int i = 0; i <= 19; i = i + 1) {
        lcd.setCursor(i, 0);
        lcd.write(byte(3));
    }
    for (int i = 1; i <= 19; i = i + 1) {
        lcd.setCursor(i, 3);
        lcd.write(byte(3));
    }
    lcd.setCursor(10, 1);
    lcd.write(byte(3));
    lcd.setCursor(10, 2);
    lcd.write(byte(3));
}

void frame3() {
    for (int i = 1; i <= 3; i = i + 1) {
        lcd.setCursor(1, i);
        lcd.write(byte(6));
    }
    for (int i = 2; i <= 17; i = i + 1) {
        lcd.setCursor(i, 1);
        lcd.write(byte(4));
    }
    for (int i = 0; i <= 3; i = i + 1) {
        lcd.setCursor(0, i);
        lcd.write(byte(3));
    }
    for (int i = 0; i <= 19; i = i + 1) {

```

```

    lcd.setCursor(i, 0);
    lcd.write(byte(3));
}
lcd.setCursor(19, 0);
lcd.write(byte(2));
lcd.setCursor(19, 1);
lcd.write(byte(2));

lcd.setCursor(17, 2);
lcd.write(byte(2));
lcd.setCursor(15, 3);
lcd.write(byte(2));
lcd.setCursor(13, 2);
lcd.write(byte(2));
lcd.setCursor(11, 3);
lcd.write(byte(2));
lcd.setCursor(9, 2);
lcd.write(byte(2));
lcd.setCursor(7, 3);
lcd.write(byte(2));
lcd.setCursor(5, 2);
lcd.write(byte(2));
lcd.setCursor(3, 3);
lcd.write(byte(2));

lcd.setCursor(17, 3);
lcd.write(byte(3));
lcd.setCursor(15, 2);
lcd.write(byte(3));
lcd.setCursor(13, 3);
lcd.write(byte(3));
lcd.setCursor(11, 2);
lcd.write(byte(3));
lcd.setCursor(9, 3);
lcd.write(byte(3));
lcd.setCursor(7, 2);
lcd.write(byte(3));
lcd.setCursor(5, 3);
lcd.write(byte(3));
lcd.setCursor(3, 2);
lcd.write(byte(3));
}

////////// Functiile pentru cele 3 nivale
void level1() {
    nrLevel = 1;

    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("      LEVEL");
    lcd.setCursor(0, 2);
    lcd.print("      1");
}

```

```

delay(1000);
lcd.clear();
lcd.setCursor(19, 3);
lcd.write(byte(3));
lcd.setCursor(xPac, yPac);
lcd.write(byte(0));
frame1();
int final = 0;
while (final == 0) {
    int state = readBtn();
    move_Pac(state);
    if (xPac == 19 && yPac == 3) {
        lcd.clear();
        you_won();
        final = 1;
    }
}
nrLevel = 0;
}

void level2() {
    nrLevel = 2;

    int final2 = 0;
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("      LEVEL");
    lcd.setCursor(0, 2);
    lcd.print("      2");
    delay(1000);
    lcd.clear();
    frame2();
    xPac = 0;
    yPac = 3;
    int c1 = 0;
    int c2 = 0;
    int c3 = 0;
    lcd.setCursor(xPac, yPac);
    lcd.write(byte(0));
    while (final2 == 0) {
        int state = readBtn();
        move_Pac(state);
        if (xPac == 0 && yPac == 0) {
            c1 = 1;
        }
        if (xPac == 19 && yPac == 0) {
            c2 = 1;
        }
        if (xPac == 19 && yPac == 3) {
            c3 = 1;
        }
        if (c1 == 1 && c2 == 1 && c3 == 1) {

```

```

    lcd.clear();
    you_won();
    final2 = 1;
}
if (xPac == 11 && yPac == 1) {
    lcd.clear();
    you_lose();
    final2 = 1;
}
if (xPac == 11 && yPac == 2) {
    lcd.clear();
    you_lose();
    final2 = 1;
}
nrLevel = 0;
}

void level13() {
    nrLevel = 3;
    int final3 = 0;
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("          LEVEL");
    lcd.setCursor(0, 2);
    lcd.print("          3");
    delay(1000);
    lcd.clear();
    frame3();
    xPac = 0;
    yPac = 3;
    lcd.setCursor(xPac, yPac);
    lcd.write(byte(0));
    while (final3 == 0) {
        int state = readBtn();
        move_Pac(state);
        if (xPac == 3 && yPac == 2) {
            lcd.clear();
            you_won();
            final3 = 1;
        }
        if (xPac == 19 && yPac == 0) {
            lcd.clear();
            you_lose();
            final3 = 1;
        }
        if (xPac == 19 && yPac == 1) {
            lcd.clear();
            you_lose();
            final3 = 1;
        }
        if (xPac == 17 && yPac == 2) {

```

```

        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 15 && yPac == 3) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 13 && yPac == 2) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 11 && yPac == 3) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 9 && yPac == 2) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 7 && yPac == 3) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 5 && yPac == 2) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
    if (xPac == 3 && yPac == 3) {
        lcd.clear();
        you_lose();
        final3 = 1;
    }
}
nrLevel = 0;
}

//////////Functii pentru afisarea mesajelor de inceput si de sfarsit de joc

```

```

void start() {
    lcd.setCursor(0, 1);
    lcd.print("      Pacman");
    lcd.setCursor(0, 1);
    for (int i = 0; i < 20; i++) {
        if (i > 0) {

```

```

    delay(200);
    lcd.setCursor(i - 1, 2);
    lcd.print(" ");
}
if (i == 0) {
    delay(200);
    lcd.setCursor(19, 2);
    lcd.print(" ");
}
if (i % 2 == 0) {
    delay(200);
    lcd.setCursor(i, 2);
    lcd.write(byte(0));
} else {
    delay(200);
    lcd.setCursor(i, 2);
    lcd.write(byte(1));
}
}

void endGame() {
    lcd.setCursor(0, 1);
    lcd.print("          END");
    lcd.setCursor(0, 2);
    lcd.print("          GAME");
    lcd.setCursor(0, 3);
    for (int i = 0; i < 20; i++) {
        if (i > 0) {
            delay(200);
            lcd.setCursor(i - 1, 3);
            lcd.print(" ");
            lcd.setCursor(i - 1, 0);
            lcd.print(" ");
        }
        if (i == 0) {
            delay(200);
            lcd.setCursor(19, 3);
            lcd.print(" ");
            lcd.setCursor(19, 0);
            lcd.print(" ");
        }
        if (i % 2 == 0) {
            delay(200);
            lcd.setCursor(i, 3);
            lcd.write(byte(0));
            lcd.setCursor(i, 0);
            lcd.write(byte(2));
        } else {
            delay(200);
            lcd.setCursor(i, 3);
            lcd.write(byte(1));
        }
    }
}

```

```

        lcd.setCursor(i, 0);
        lcd.write(byte(2));
    }
}
}

//////////Functie pentru luarea deciziei in functie de butonul apasat

void move_Pac(int state) {
    switch (state) {
        case rightBtn:
            v = verify(xPac + 1, yPac);
            if (v == 1) {
                xPac = xPac + 1;
                move_rigth(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case upBtn:
            v = verify(xPac, yPac - 1);
            if (v == 1) {
                yPac = yPac - 1;
                move_up(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case downBtn:
            v = verify(xPac, yPac + 1);
            if (v == 1) {
                yPac = yPac + 1;
                move_down(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        case leftBtn:
            v = verify(xPac - 1, yPac);
            if (v == 1) {
                xPac = xPac - 1;
                move_left(xPac, yPac, pacState);
                pacState = ~pacState;
                delay(300);
            }
            break;
        default:
            // statements
            break;
    }
}

```

```

////////// functiile pentru mutarea Pacman-ului in SUS,JOS,STANGA,DREAPTA

void move_down(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x, y - 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x, y - 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(0));
    }
}

void move_up(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x, y + 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x, y + 1);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(0));
    }
}

void move_left(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x + 1, y);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x + 1, y);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);

```

```

        lcd.write(byte(0));
    }
}

void move_right(int x, int y, int s) {
    if (s == 0) {
        delay(200);
        lcd.setCursor(x - 1, y);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(1));
    } else {
        delay(200);
        lcd.setCursor(x - 1, y);
        lcd.print(" ");
        delay(200);
        lcd.setCursor(x, y);
        lcd.write(byte(0));
    }
}

```

//////////verificarea ca pacman-ul sa ramana in matricea de 20x4 a lcd-ului

```

int verify(int x, int y) {
    if ((x <= 19 && x >= 0) && (y <= 3 && y >= 0)) {
        int ver;
        if (nrLevel == 1) {
            ver = verifyFrame1(x, y);
        }
        if (nrLevel == 2) {
            ver = verifyFrame2(x, y);
        }
        if (nrLevel == 3) {
            ver = verifyFrame3(x, y);
        }
        if (ver == 1) {
            return 1;
        }
    } else {
        return 0;
    }
}

```

//////////Verificarea frame-urilor astfel incat pacmanul sa nu poata intra in ziduri

```

int verifyFrame1(int x, int y) {
    for (int i = 6; i <= 19; i = i + 1) {
        if (x == i && y == 1) {
            return 0;
        }
    }
}

```

```

for (int i = 6; i <= 19; i = i + 1) {
    if (x == i && y == 2) {
        return 0;
    }
}

return 1;
}

int verifyFrame2(int x, int y) {
    for (int i = 0; i <= 9; i = i + 1) {
        if (x == i && y == 1) {
            return 0;
        }
    }
    for (int i = 0; i <= 9; i = i + 1) {
        if (x == i && y == 2) {
            return 0;
        }
    }
    for (int i = 12; i <= 19; i = i + 1) {
        if (x == i && y == 1) {
            return 0;
        }
    }
    for (int i = 12; i <= 19; i = i + 1) {
        if (x == i && y == 2) {
            return 0;
        }
    }
}

return 1;
}

int verifyFrame3(int x, int y) {
    for (int i = 1; i <= 3; i = i + 1) {
        if (x == 1 && y == i) {
            return 0;
        }
    }
    for (int i = 2; i <= 17; i = i + 1) {
        if (x == i && y == 1) {
            return 0;
        }
    }
}

return 1;
}

////////// Citirea butoanelor de pe Bread Board pentru controlarea Pacman-ului

int readBtn() {
    buttonState1 = digitalRead(leftBtn);
}

```

```

buttonState2 = digitalRead(upBtn);
buttonState3 = digitalRead(rightBtn);
buttonState4 = digitalRead(downBtn);

if (buttonState1 == LOW) {
    return leftBtn;
}
if (buttonState2 == LOW) {
    return upBtn;
}
if (buttonState3 == LOW) {
    return rightBtn;
}
if (buttonState4 == LOW) {
    return downBtn;
}
}

```

////////// Functiile pentru castig si pentru pierdut

```

void you_lose() {
    lcd.setCursor(0, 1);
    lcd.print("          YOU");
    lcd.setCursor(0, 2);
    lcd.print("          LOSE");
    lose_song();
}

```

```

void you_won() {
    lcd.setCursor(0, 1);
    lcd.print("          YOU");
    lcd.setCursor(0, 2);
    lcd.print("          WON");
    win_song();
}

```

////////// Cantecele pentru Buzzer

```

void win_song() {
    tone(buzzer, 420, 200);
    delay(500);
    tone(buzzer, 440, 200);
    delay(500);
    tone(buzzer, 490, 270);
    delay(300);
    tone(buzzer, 520, 200);
    delay(300);
}

```

```

void lose_song() {
    tone(buzzer, 190, 230);
    delay(500);
}

```

```
tone(buzzer, 100, 230);
delay(500);
tone(buzzer, 50, 270);
delay(300);
}
```