

Dobre Emilia Iliana 333CB

# Snake Game □

## Introducere

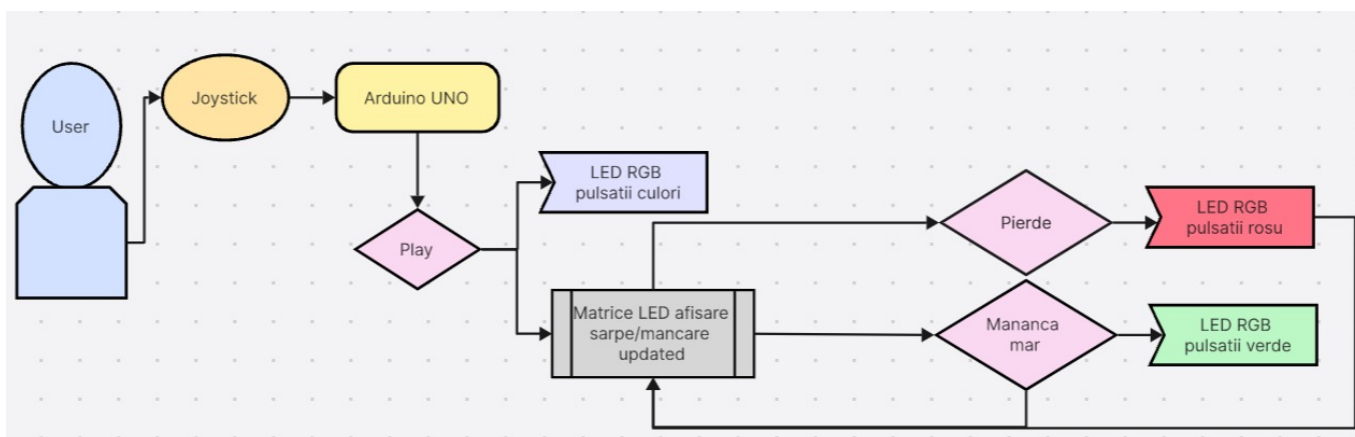
Proiectul meu, Snake Game, este o reinterpretare a clasicului joc Snake, conceput să ofere divertisment și nostalgie. Jucătorul controlează un șarpe într-un mediu închis, cu scopul de a mânca bucăți de “hrană” care apar aleatoriu pe o matrice LED de 8x8. Cu fiecare bucată de hrană consumată, șarpele crește în lungime, iar jocul devine progresiv mai dificil. Obiectivul este să acumulezi cât mai multe puncte posibil, evitând coliziunea cu propriul corp al șarpelui.

Inspirat de jocurile arcade clasice, acest proiect combină nostalgia jocurilor retro cu tehnologia modernă Arduino, oferind o experiență simplă, dar captivantă, accesibilă oricărui entuziast de jocuri sau hobbyist de electronică.

## Descriere generală

Proiectul este construit pe o platformă Arduino UNO și utilizează un modul MAX7219 Dot Matrix 8x8 pentru afișarea vizuală a jocului. Controlul șarpelui se face prin intermediul unui joystick care permite mișcarea în sus, jos, stânga sau dreapta.

Șarpele în sine este reprezentat pe matricea de LED-uri ca o serie de lumini aprinse consecutive, care corespund lungimii sale. Lungimea șarpelui crește de fiecare dată când consumă o bucată de “hrană”, care este reprezentată de un LED aprins într-o locație aleatoare pe matrice. Pe măsură ce jucătorul dirijează șarpele prin intermediul joystick-ului, LED-urile care reprezintă corpul șarpelui se aprind și se sting în secvență, reconfigurându-se pentru a ilustra mișcarea acestuia pe matrice. Astfel, când șarpele se mișcă, LED-urile vechi se sting și cele noi se aprind în direcția mișcării. De asemenea, pentru acest proiect exista un LED RGB care va indica starea jocului (verde când mănâncă, roșu când se termină jocul, și alte culori pe parcursul mișcării).



# Hardware Design

## Componente folosite și rolul lor în proiect:

- Arduino UNO: Placa principală care va rula codul jocului.
- Modul MAX7219 Dot Matrix 8x8: Va afișa starea jocului, inclusiv poziția șarpelui și mâncarea.
- Joystick Analogic: Permite utilizatorului să controleze direcția șarpelui.
- LED RGB: Indică diferite stări ale jocului prin schimbarea culorii.
- Rezistențe: Folosite pentru a limita curentul prin LED-ul RGB.
- Breadboard
- Cabluri conectare

## Pini folosiți/motivul utilizării acestora:

Arduino Uno are:

- **pini digitali(0-13)** care pot fi folosiți pentru a citi sau scrie stări digitale (HIGH sau LOW).
- **pini Digitali cu Funcție PWM** care pot genera semnale PWM (Pulse Width Modulation).
- **pini analogici (A0-A5)** care sunt folosiți pentru citirea valorilor analogice.

## Matricea LED:

VCC ( Voltage Common Collector) - conectat la pinul de 5V al Arduino asigură alimentarea necesară funcționării matricei de LED-uri.  
GND ( Ground) - punctul de referință de tensiune zero în circuit, astfel GND al matricei LED MAX72XX este conectat la pinul GND al Arduino  
DIN (Data In) - conectat la pinul digital 12 de pe Arduino: folosit pentru a trimite date către matrice.  
CS (Chip Select) - conectat la pinul digital 11 de pe Arduino: folosit pentru a selecta matricea activă.  
CLK (Clock) - conectat la pinul digital 10 de pe Arduino: folosit pentru a trimite semnale de sincronizare către matrice.

**Joystick analogic:** Un joystick analogic are două potențiometre încorporate, unul pentru axa X și unul pentru axa Y. Fiecare potențiometru trimite o valoare de tensiune care variază în funcție de poziția joystick-ului. Această valoare de tensiune este citită de pini analogici de pe Arduino.

VCC - conectat la pinul 5V al Arduino, alimentare joystick  
GND - conectat la pinul GND al Arduino.  
VRx - conectat la pinul analogic A3 al Arduino pentru a permite Arduino să citească valoarea analogică corespunzătoare poziției pe axa X.  
VRY - conectat la pinul analogic A4 al Arduino pentru a permite Arduino să citească valoarea analogică corespunzătoare poziției pe axa Y.

## LED RGB:

Catod comun: conectat la pinul GND al Arduino pentru a oferi o referință de tensiune zero comună pentru toate cele trei LED-uri interne.

Roșu: pin 7  
Verde: pin 6  
Albastru: pin 5  
Rezistențe de 220Ω: rotejează LED-ul de curentul excesiv  
Pini digitali 5,6,7 controlează intensitatea lumininilor folosind semnale PWM de la Arduino.

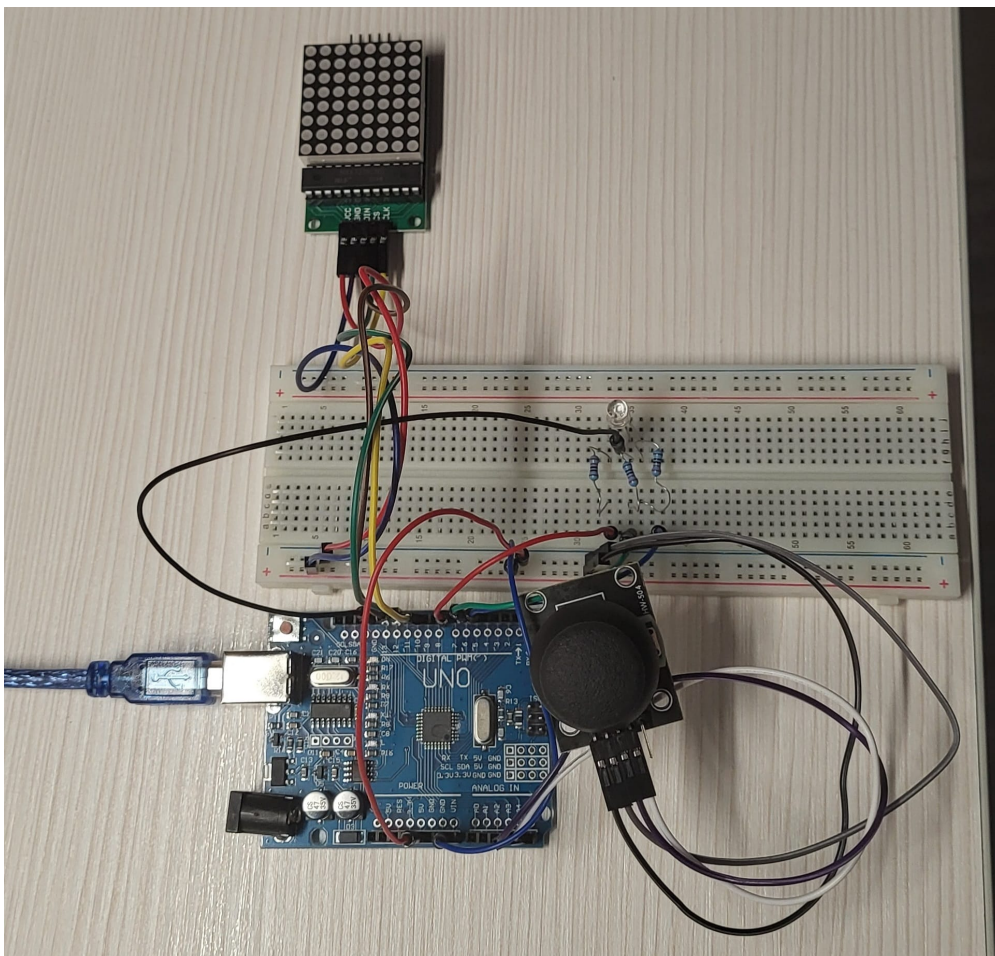
### Schema hardware:

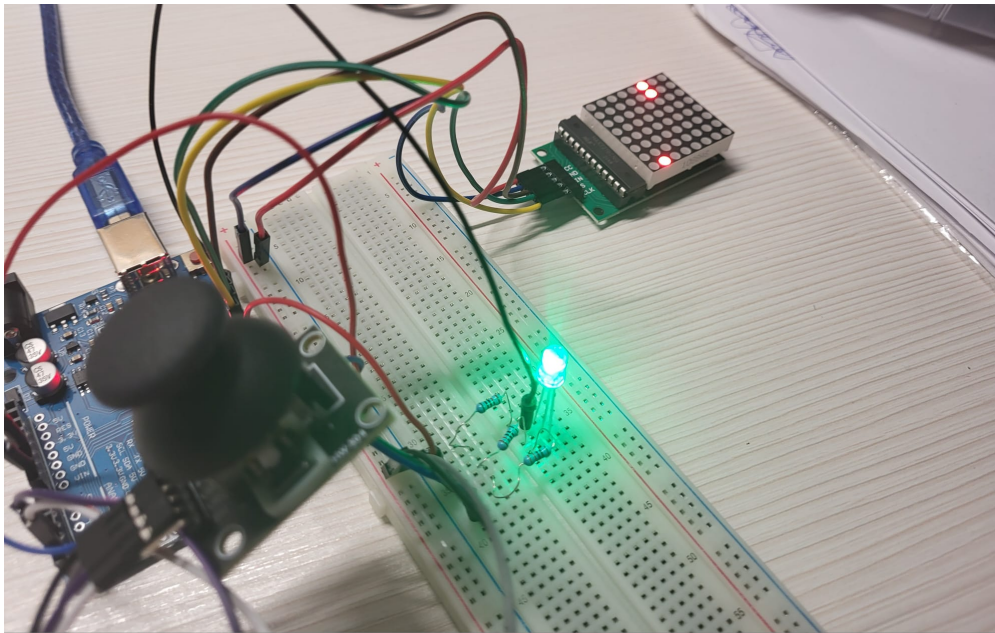


### Schema electrica:



### Testare functionalitate:





## Software Design

### Mediu de dezvoltare:

- **Arduino IDE** care oferă o platformă accesibilă și ușor de utilizat, cu suport pentru compilarea și încărcarea codului direct pe placa Arduino.

### Biblioteci Utilizate

- **LedControl**: Aceasta bibliotecă este esențială pentru controlul modulului MAX7219 Dot Matrix. Permite manipularea ușoară a LED-urilor pe matrice, inclusiv aprinderea și stingerea acestora, ceea ce este crucial pentru vizualizarea șarpelui și a mâncării în joc.
- **Standard Arduino Libraries**: Bibliotecile standard ale Arduino (`analogRead`, `pinMode`, `digitalWrite`, `analogWrite`, etc.) sunt utilizate pentru a interacționa cu joystick-ul și LED-ul RGB.

### Elementul de noutate al proiectului

Elementul de noutate al proiectului constă în integrarea efectelor vizuale cu LED-uri RGB, care oferă feedback vizual în funcție de acțiunile jucătorului. De exemplu:

- LED-ul RGB pâlpâie verde când șarpele mănâncă un măr.
- LED-ul RGB devine roșu când jocul se termină.
- În timpul jocului, LED-ul RGB își schimbă culoarea în mod constant pentru a adăuga un efect vizual dinamic.

### Funcționalități din laborator

- **GPIO**: Pinii GPIO sunt utilizați pentru a citi valorile de la joystick și pentru a controla starea LED-urilor RGB.
- **PWM**: Pulse Width Modulation este utilizat pentru a controla intensitatea diferitelor culori ale LED-ului RGB.
- **ADC**: Conversia Analog-Digitală este utilizată pentru a citi poziția joystick-ului.

- **SPI:** Interfața SPI este utilizată pentru a comunica cu matricea LED MAX72XX, permițând actualizarea rapidă a ecranului.

### Explicația scheletului proiectului

- Setup: Funcția `setup()` inițializează comunicarea cu matricea LED, configurează pinii pentru joystick și LED-ul RGB, și setează seed-ul pentru generatorul de numere aleatoare.
- Loop: Funcția `loop()` gestionează citirea intrărilor de la joystick, actualizarea poziției șarpelui, schimbarea culorilor LED-ului RGB și redarea graficii pe matricea LED.
- Update: Funcția `Update()` calculează noua poziție a capului șarpelui, verifică coliziunile și consumul mărului, și actualizează starea șarpelui.
- Render: Funcția `Render()` afișează starea curentă a jocului pe matricea LED.
- `changeColor`: Funcția `changeColor()` schimbă culoarea LED-ului RGB în mod constant în timpul jocului.
- `eatAppleEffect`: Funcția `eatAppleEffect()` face LED-ul RGB să pâlpâie verde atunci când șarpele mănâncă un măr.
- `gameOver`: Funcția `gameOver()` schimbă culoarea LED-ului RGB în roșu atunci când jocul se termină.

### Algoritmi și Structuri Propuse

#### • Algoritm de Mișcare a Șarpelui:

Calculez noua poziție a capului șarpelui (`newHead`) pe baza direcției curente. Implementez logica de "wrap around" pentru a face șarpele să apară pe partea opusă a matricei LED dacă depășește marginile. Verific dacă noua poziție a capului șarpelui coincide cu oricare dintre segmentele corpului său. Dacă da, jocul se termină. Setez bit-urile corespunzătoare pentru fiecare segment al corpului șarpelui și pentru măr în array-ul `pic`, care reprezintă starea matricei LED.

- Logica matricei LED și bit-shifting:

Matricea LED este reprezentată de un array de 8 elemente (`byte pic[8]`), fiecare element reprezentând o linie a matricei LED de 8×8. Iecare bit dintr-un byte reprezintă un LED din linia respectivă:

128 în binar este 10000000.

128 » col deplasează bit-ul 1 spre dreapta cu numărul de poziții specificat de col (de exemplu, dacă col este 3, rezultatul este 00010000).

Operatorul `|= (128 » col)` este utilizat pentru a seta bit-ul corespunzător în rândul specificat din matricea LED, aprinzând astfel LED-ul la poziția dorită.

- **Generarea Aleatoare a Mâncării:** Un mecanism pentru plasarea aleatorie a mâncării pe matrice, asigurându-se că aceasta nu apare pe corpul șarpelui.

### Calibrarea elementelor de senzorială

- Joystick: Valorile citite de la joystick au fost calibrate pentru a determina pragurile pentru fiecare direcție. Aceste praguri au fost ajustate în cod pentru a asigura o reacție precisă la mișcările jucătorului.
- LED RGB: Intensitatea fiecărei culori a fost ajustată folosind PWM pentru a obține efecte vizuale clare și distincte.

## Optimizări

\* Utilizarea SPI: Interfața SPI permite actualizarea rapidă a matricei LED, asigurând o afișare fluidă a graficii jocului. \* Reducerea utilizării delay(): În afară de efectele de pâlpâire ale LED-ului RGB, utilizarea delay() a fost minimizată pentru a nu bloca execuția altor funcții. \* Structurarea codului: Codul a fost structurat în funcții distincte pentru a simplifica mentenanța și extinderea acestuia.

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

Github: <https://github.com/EmiDobre/Snake-Game-Arduino-Uno>

## Jurnal

- 24.04.2024: Alegere tema proiect
- 01.05.2024: Cumparare componente
- 05.05.2024: Scriere documentatie
- 17.05.2024: Creare repo github
- 17.05.2024: Adaugare sectiuni lipsa
- 19.05.2024: Creare schema electrica
- 20.05.2024: Testare conectivitate componente
- 22-25.05.2024: Cod
- 26.05.2024: Etapa software

TODO:

- Finalizare hardware, incepere software
- Research, scriere cod
- Adaugare concluzii in documentatie



## Bibliografie/Resurse

### Resurse:

- circuit:

<https://www.circuito.io/>

<https://easyeda.com/>

- set up arduino:

<https://www.youtube.com/watch?v=JnJKX5J0Cc&list=PLwWF-ICTWmB7-b9bsE3UcQzz-7ipI5tbR>

- led: lgb: <https://www.instructables.com/RGB-LED-With-Arduino-Uno-R3/>

- led control: <https://www.youtube.com/watch?v=NEXjteud3is>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/emilia\\_iliana.dobre](http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/emilia_iliana.dobre) 

Last update: **2024/05/25 22:48**