

# Snake Game

## Introducere

- Acest proiect reprezintă o implementare a clasicului joc Snake pe o placă Arduino.
- Scopul jocului este de a controla un șarpe care se mișcă pe un ecran și de a mânca bucăți de hrană pentru a crește lungimea șarpelui, evitând coliziunile cu marginile ecranului sau cu corpul său.
- Ideea de la care am porit a fost dorința de a reinterpretă un joc al copilăriei.
- Consider că acest joc este util pentru a arăta cum se pot îmbina conceptele învățate la laborator pentru a implementa un joc mai complex.

## Descriere generală

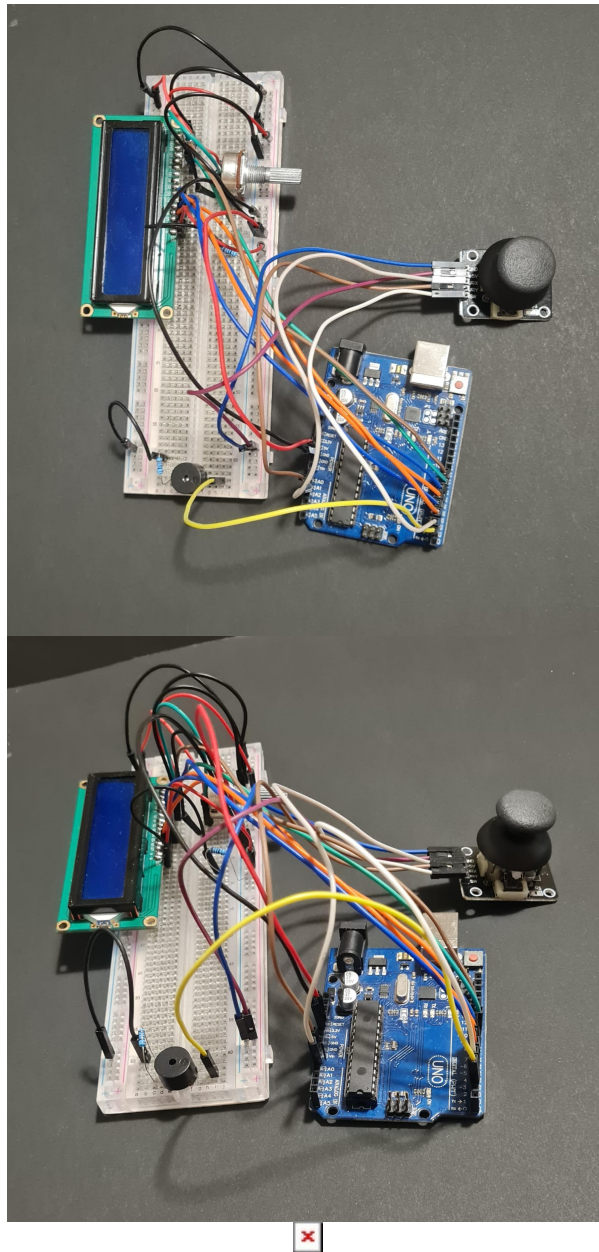
Cu ajutorul unui breadboard și al firelor, cele patru componente: plăcuța Arduino, joystick-ul, matricea cu leduri și buzzer-ul sunt conectate. Jucătorul va controla snake-ul cu ajutorul joystick-ului și va încerca să strângă cât mai multe puncte prin a mânca luminițele de pe traseu. Buzzer-ul va scoate un sunet la pierderea jocului care se va întâmpla dacă șarpele atinge marginile matricei.



## Hardware Design

Componente hardware:

1. Matrice de leduri
2. Arduino Uno
3. Buzzer
4. Breadboard
5. Fire
6. Joystick
7. MAX7219 Driver
8. Potentiometru
9. Ecran LCD



## Software Design

- Am implementat jocul în Arduino IDE
- Am utilizat librăriile LedControl, LiquidCrystal, EEPROM
- Am utilizat o structura pentru șarpele din joc, care salveaza potitiile capului, corpului, lungimea sa si directia.
- Am utilizat o structura pentru mărul pe care trebuie să-l mănânce pentru a putea crește. După ce șarpele ajunge să ia un măr altul va apărea într-un alt loc random din matrice.

### Actualizarea și Mișcarea Șarpelui:

- Direcția șarpelui este controlată de valorile citite de la joystick (varXPin și varYPin). Dacă joystick-ul este mișcat într-o direcție, direcția șarpelui se schimbă corespunzător. Mișcarea șarpelui este realizată prin actualizarea poziției capului (head) pe baza direcției curente (dir). Dacă capul șarpelui se mută peste marginea matricei de LED-uri, poziția sa este ajustată pentru a apărea pe partea opusă (efect de teleportare).

- Coliziunea cu sine: Verifică dacă coordonatele capului șarpelui coincid cu oricare parte a corpului său. Dacă da, jocul se resetează, iar jucătorul pierde o viață.
- Coliziunea cu mărul: Dacă capul șarpelui ajunge la poziția mărului, șarpele crește în lungime, scorul este incrementat și un nou măr este generat la poziții aleatorii.
- Matricea de LED-uri este actualizată pentru a afișa noua poziție a șarpelui și a mărului. Șarpele este desenat folosind array-ul pic, iar mărul este făcut să clipească pentru a fi mai vizibil.

Meniuri și Interfață:

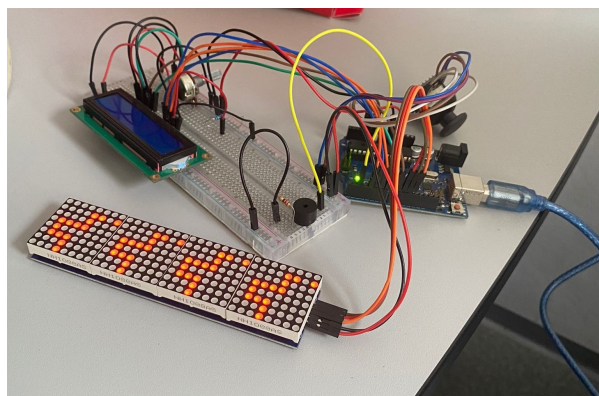
- Interfața jocului este realizată folosind un ecran LCD. Meniurile sunt navigate utilizând butonul joystick-ului. Există mai multe meniuri și submeniuri pentru opțiuni precum începerea jocului, setările, etc.

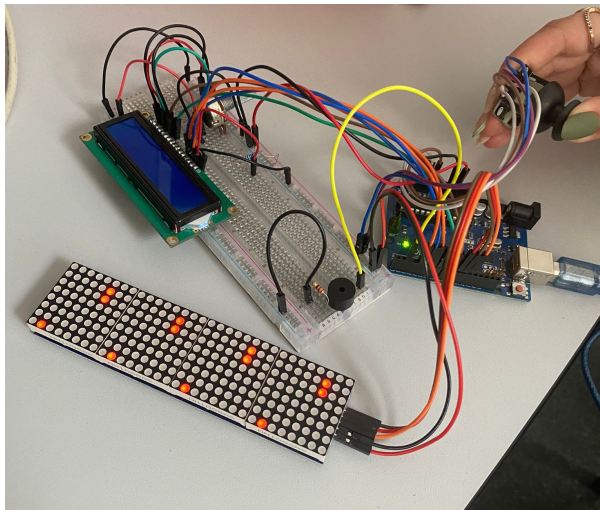
Funcții Importante:

1. playGame(): Gestionează logica principală a jocului.
2. showIntro(): Afișează introducerea jocului.
3. showMenu(): Afișează meniul principal și permite navigarea prin opțiuni.
4. Update(): Actualizează starea șarpelui și a mărului.
5. Render(): Redă (desenează) jocul pe matricea de LED-uri.
6. calculateDeltaTime(): Calculează timpul trecut între cadre pentru a controla viteza jocului.
7. removeFirst(): Elimină prima parte a corpului șarpelui pentru a simula mișcarea sa înainte.

## Rezultate Obținute

Pe matricea de leduri apare un model initial. Apăsarea joystick-ului duce la începerea jocului. Un snake format inițial din două cercuri luminoase se deplasează pe matrice și este controlat de către jucător cu ajutorul joystick-ului. La începutul jocului se poate auzi un cântecel, un sunet scurt se aude la mâncarea merelor, apare câte unul pe rând în locuri random. Șarpele crește odată cu mâncarea punctelor luminoase și își pierde viața dacă își atinge propriul corp. La pierderea jocului se va auzi un sunet mai lung și mai puternic.





## Concluzii

Mi-a făcut plăcere să lucrez cu ceva nou și să pot implementa un joc clasic care mi-a plăcut foarte mult în copilărie.

## Download

[costea\\_maria\\_cristina\\_pm.zip](#)

## Jurnal

2.05 - Alegerea temei proiectului

9.05 - Prezentarea pieselor pe care urma sa le utilizez

12.05 - Legarea pieselor hardware

13.05 - Intocmirea documentatiei pentru partea de hardware

16.05 - Prezentarea celui de al-doilea milestone

19.05 - 21.05 - Scrierea codului

23.05 - Prezentarea ultimului milestone

## Bibliografie/Resurse

- <https://docs.arduino.cc/built-in-examples/usb/JoystickMouseControl/>
- <https://docs.arduino.cc/built-in-examples/display/RowColumnScanning/>
- <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- <https://projecthub.arduino.cc/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-cf4191>
- <https://docs.arduino.cc/built-in-examples/display/RowColumnScanning/>
- <https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoToBreadboard/>
- <https://www.tinkercad.com/dashboard>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/alucaci/maria.costea2604>



Last update: **2024/05/23 15:12**