

Joc Snake

Introducere

În cadrul proiectului mi-am propus să realizez un joc clasic și captivant: Snake. Pentru acesta, voi utiliza o placă Arduino, un display LCD, joystick și buzzer. Scopul este de a face nu doar un joc, ci și o experiență interactivă, care să demonstreze capacitatea în programare și interacțiunea cu hardware-ul.

Descriere generală

Scopul jocului este simplu: controlând un șarpe, jucătorul trebuie să mănânce cât mai multă hrană posibilă și să evite coliziunea.

Pentru a aduce acest joc la viață, voi folosi o placă Arduino ca bază pentru controlul jocului. Voi conecta un display LCD care va afișa ecranul de joc, joystick pentru mișcarea jucătorului și un Buzzer pentru a atenționa anumite interacțiuni.

Astfel, prin acest proiect, voi explora aspecte ale programării embedded, interacțiunii cu perifericele și controlului hardware-ului. De asemenea, oferă o experiență practică și distractivă, care să stimuleze creativitatea și pasiunea pentru tehnologie.

Hardware Design

- Display LCD matrice 8×8
- Joystick
- Placă Arduino Uno R3
- Buzzer
- Breadboard

Rolul pieselor și pinii folosiți

1. Placă Arduino Uno R3:

Rol: Controlerul principal care coordonează toate componentele și logica jocului "Snake".

2. Joystick

Rol: Interfața de control a jocului, folosită pentru a mișca șarpele în direcțiile dorite.

Pini:

```
## GND (joystick) -> GND (breadboard): Asigură conexiunea la masa comună.  
## +5V (joystick) -> +3.3V (breadboard): Alimentează joystick-ul.  
## VRx (joystick) -> A0 (Arduino): Citește valoarea de pe axa X.  
## VRy (joystick) -> A1 (Arduino): Citește valoarea de pe axa Y.  
## SW (joystick) -> D2 (Arduino): Citește starea butonului joystick-ului.
```

3. Buzzer

Rol: Emite sunete pentru a oferi feedback auditiv, cum ar fi atunci când șarpele mănâncă sau se lovește de un obstacol.

Pini:

```
## Pozitiv (+) (buzzer) -> D3 (Arduino): Controlează buzzer-ul.  
## Negativ (-) (buzzer) -> GND (breadboard): Asigură conexiunea la masa comună.
```

4. Display LCD matrice 8x8

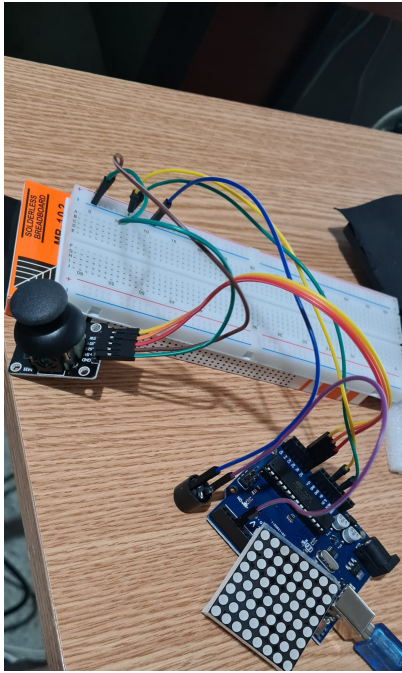
Rol: Ecranul pe care se va afișa jocul "Snake"

Pini:

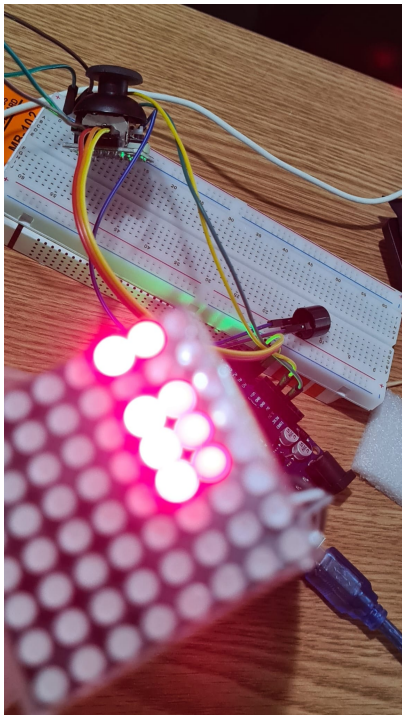
```
## VCC (MAX7219) -> 5V (Arduino): Alimentează driver-ul MAX7219.  
## GND (MAX7219) -> GND (Arduino): Asigură conexiunea la masa comună.  
## DIN (MAX7219) -> D11 (Arduino): Linie de date pentru comunicare SPI.  
## CS (MAX7219) -> D10 (Arduino): Linie de selectare a chip-ului pentru comunicare SPI.  
## CLK (MAX7219) -> D13 (Arduino): Linie de ceas pentru comunicare SPI.
```

Hardware + Schema electrică

Hardware asamblat



LCD Aprins



Miscare Joystick



Schema Electrică



Software Design

Stadiul Final

În stadiul actual, jocul permite utilizatorilor să controleze un șarpe cu ajutorul unui joystick, să mănânce fructe pentru a crește în dimensiune și să evite coliziunile cu pereții sau cu propriul corp. Jocul include următoarele funcționalități:

- Controlul Șarpelui:** Șarpele este controlat printr-un joystick, iar direcția sa se schimbă în funcție de mișcările joystick-ului.
- Creșterea Dimensiunii:** Șarpele crește în dimensiune după ce mănâncă un anumit număr de fructe.
- Generarea Fructelor:** Fructele apar în poziții aleatorii pe matricea de LED-uri, asigurându-se că nu se suprapun cu șarpele.
- Coliziuni:** Jocul se termină dacă șarpele se lovește de pereți sau de propriul corp.
- Sunet:** Buzzerul emite sunete pentru diferite acțiuni, cum ar fi începutul jocului, creșterea în dimensiune și "game over".
- Afișare și Clipire:** Șarpele clipește pe matricea de LED-uri pentru a indica "game over".

Biblioteci

1. LedControl:

Motivație: Biblioteca LedControl este utilizată pentru a controla matricea de LED-uri 8x8. Aceasta permite o interfațare simplă și eficientă cu hardware-ul LED.

Funcționalitate: Permite activarea și dezactivarea individuală a LED-urilor și setarea intensității luminoase.

2. Arduino:

Motivație: Biblioteca de bază Arduino oferă funcționalitățile necesare pentru citirea valorilor analogice de la joystick și generarea de tonuri pentru buzzer.

Funcționalitate: Permite configurarea pinilor, citirea valorilor analogice și digitale, și generarea de tonuri sonore.

Schelet proiect

setup(): Inițializează hardware-ul (joystick, matrice de LED-uri, buzzer) și setează starea inițială a jocului.

loop(): Monitorizează starea jocului și actualizează poziția șarpelui, verifică coliziunile și redă sunetele corespunzătoare.

Funcții Auxiliare:

1. readJoystick()

Descriere: Citește valorile analogice de la joystick și actualizează direcția șarpelui în funcție de aceste valori.

Rol: Permite controlul direcției șarpelui prin mișcările joystick-ului.

2. moveSnake()

Descriere: Actualizează poziția șarpelui în funcție de direcția curentă.

Rol: Mută șarpele în direcția specificată, actualizând pozițiile fiecărei părți a corpului său.

3. checkCollision()

Descriere: Verifică dacă șarpele a colizionat cu pereții, cu propriul corp sau a mâncat fructul.

Rol: Determină dacă jocul s-a terminat și dacă șarpele a mâncat un fruct, actualizând lungimea acestuia și generând un nou fruct.

4. generateFruit()

Descriere: Generează coordonate aleatorii pentru un nou fruct, asigurându-se că acesta nu se suprapune cu corpul șarpelui.

Rol: Plasează corect fructul pe matricea de LED-uri.

5. draw()

Descriere: Actualizează afișajul matricei de LED-uri pentru a reflecta pozițiile șarpelui și ale fructului.

Rol: Desenează șarpele și fructul pe matricea de LED-uri.

6. displayGameOver()

Descriere: Afișează mesajul de "game over" și face ca șarpele să clipească.

Rol: Indică sfârșitul jocului și oferă feedback vizual.

7. playBuzzerTone(int frequency, int duration)

Descriere: Redă un ton cu o anumită frecvență și durată folosind buzzerul.

Rol: Oferă feedback auditiv pentru diferite acțiuni din joc.

8. smoothAnalogRead(int pin)

Descriere: Citește valorile analogice de la un pin specificat și returnează media mai multor citiri pentru a obține o valoare mai stabilă.

Rol: Reduce fluctuațiile citirilor analogice de la joystick.

Interacțiuni & Validare funcționalități

A. Interacțiune

1. Citirea Joystick-ului: Citirea valorilor analogice pentru a determina direcția șarpelui.
2. Actualizarea Stării Jocului: Mișcarea șarpelui și generarea de fructe.
3. Afișarea pe Matricea de LED-uri: Actualizarea LED-urilor pentru a reflecta pozițiile șarpelui și ale fructului.
4. Feedback Auditiv: Generarea de tonuri pentru evenimente specifice (start, creștere, game over).

B. Validare

1. Testare Unit: Fiecare componentă (joystick, matrice de LED-uri, buzzer) a fost testată individual pentru a se asigura că funcționează corect.
2. Testare Integrată: Testarea completă a jocului pentru a verifica că toate funcționalitățile interacționează corect și că starea jocului este gestionată corespunzător.

Calibrare elemente de senzorică (Joystick)

- 1. Citirea Valorilor Analogice:** Utilizarea funcției `analogRead()` pentru a citi valorile de la VRx și VRy.
- 2. Zonă Moartă:** Implementarea unei zone “moarte” pentru a preveni mișcările nedorite atunci când joystick-ul este în poziția centrală.
- 3. Smoothing:** Utilizarea unui filtru de mediere pentru a stabiliza valorile citite de la joystick (`smoothAnalogRead()`).

DEMO

<https://www.youtube.com/watch?v=J2fpi5qFztg>

Download

[vitregu_rares_344c1.zip](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/vitregu.rares>



Last update: **2024/05/26 01:31**