

3



Universitatea de Vest Timișoara

Facultatea de Matematică și Informatică

Departamentul Informatică Română

---

## Arduino Snake Game

---

*Autori:*

Alexandru Sebastian *Tufis-Schwartz*

Sebastian Ioan *Suciu*

Ştefania Denisa *Ionescu*

Vlad *Brudaşcă*

10 iulie 2017

## **Rezumat**

Lucrarea constă în implementarea jocului Snake cu suport fizic. Jocul constă într-o matrice de  $8 \times 8$  din leduri RGB WS2812b și un controller ca suport input/output. Componentele sunt controlate cu Arduino. Jocul a fost implementat în limbajul C++ și Arduino.

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>3</b>
1.1	Motivație . . . . .	3
1.2	Componente . . . . .	3
<b>2</b>	<b>Construire hardware</b>	<b>3</b>
2.1	Pregătirea tablei . . . . .	3
2.2	Cablare LED-uri . . . . .	4
2.3	Conectare LED-uri . . . . .	4
2.4	Asamblare carcăsă . . . . .	6
2.5	Construirea matricei . . . . .	6
2.6	Construirea controllerului . . . . .	8
<b>3</b>	<b>Implementare cod</b>	<b>9</b>
<b>4</b>	<b>Rezultate</b>	<b>10</b>
<b>5</b>	<b>Concluzii</b>	<b>11</b>

# 1 Introducere

## 1.1 Motivație

Prin acest proiect am dorit să implementăm un joc în C++ pe un suport hardware construit manual, respectiv un controller(input) și o matrice de LED-uri RGB, prin intermediul Arduino. Jocul Snake a fost ales deoarece implementarea a fost potrivită pentru suportul hardware construit.

## 1.2 Componente

Componentele necesare:

1. 65 LED-uri WS2812b
2. Arduino UNO
3. Tablă de lemn
4. Sticlă de 2.5mm acrilică opal
5. Cabluri rigide de  $1.5 \times 1.5$
6. Cabluri flexibile de  $0.75 \times 0.75$
7. Sticlă acrilică de 2.5mm
8. Butoane controller  $x4$
9. Cabluri de legătură

# 2 Construire hardware

## 2.1 Pregătirea tablei

Primul pas constă în pregătirea tablei. A fost folosită o tablă de șah ca suport. A fost desenat un  $X$  în interiorul fiecărui patrat pentru a-i găsi mijlocul. LED-ul a fost plasat în mijlocul trasat. În jurul LED-ului au fost date trei orificii în partea de jos a acestuia de 2mm și o cavitate de aceeași dimensiune în partea de sus a sa.



Figura 1: LED-ul plasat pe tabla de șah cu cele 4 orificii.

## 2.2 Cablare LED-uri

Cele trei cavități din partea de jos a LED-ului sunt *GND*, *DATA IN* și *5V*, iar orificiu situat superior este cablul *DATA OUT*. Pe partea posterioară a LED-ului este marcată o căte o săgeată ce indică direcția corespunzătoare.



Figura 2: Modul de asamblare a cablurilor pe LED.

## 2.3 Conectare LED-uri

LED-ul se conectează pe partea din spate a tablei. Sunt necesare nouă bucăți de cablu rigid, câte unul între fiecare rând de LED-uri și la începutul și sfârșitul matricei, respectiv cinci pentru *GND* și patru pentru *V5*. *GND*-ul LED-ului se conectează la cablul *GND*, iar *V5*-ul la cablul *V5*.

Primul *DATA IN* se conectează la Arduino și *DATA OUT* la următorul LED. Ultimul LED *DATA OUT* al rândului este conectat la primul *DATA IN* al următorului rând.

La final se conectează firele *GND* și *5V* împreună și se conectează la pinii corespunzători din Arduino. S-a folosit de asemenea un pistol de lipit pentru izolarea cablurilor.

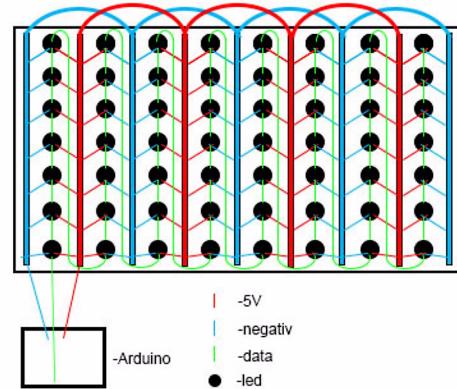


Figura 3: Schița asamblării utilizată.



Figura 4: Asamblarea finală a cablurilor.

## 2.4 Asamblare carcasă

Pentru asamblarea carcasei s-au folosit:

1. Bucată de lemn pentru margini  $x4$



Figura 5: Marginile carcasei asamblate.

2. O bucată de lemn pentru spatele carcasei
3. Sticlă acrilică pentru display



(a) Spatele carcasei.



(b) Display-ul jocului.

Figura 6: Fețe carcasă.

## 2.5 Construirea matricei

Pentru asamblarea finală a matricei au fost necesare:

1. Cincisprăzece stinghii pentru delimitarea ariei ledurilor



Figura 7: Stinghii crestate pentru delimitare LED.

2. Patru stinghii pentru delimitarea marginii matricei



Figura 8: Stinghii simple delimitare margini.

Asamblarea finală a matricei se poate observa în figura de mai jos.

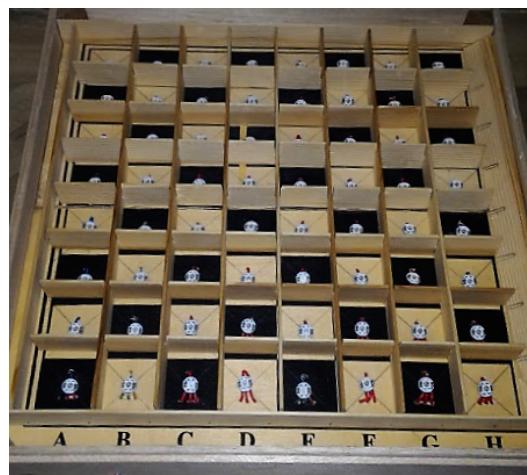


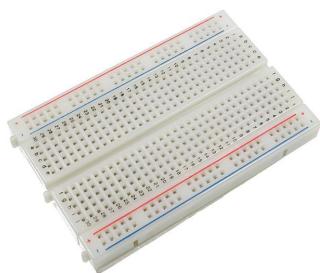
Figura 9: Matrice 8x8 WS2812 LED finalizată.

## 2.6 Construirea controllerului

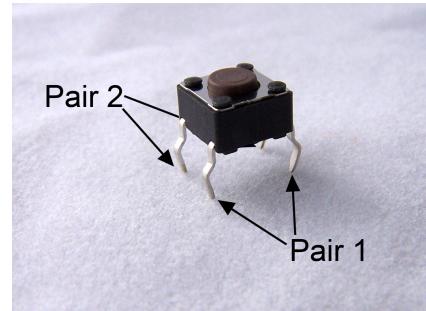
Controllerul jocului de Snake constă în patru butoane(sus, jos, stânga, dreapta), după cum se poate observa în Figura 12, ce trimit către Arduino semnale interpretate ca direcție.

Componentele controllerului sunt:

1. Breadboard
2. Push Button asamblat pe Breadboard *x4*



(a) Breadboard.



(b) Push Button.

Figura 10: Componente controller.

3. Cabluri pentru conectarea la Arduino

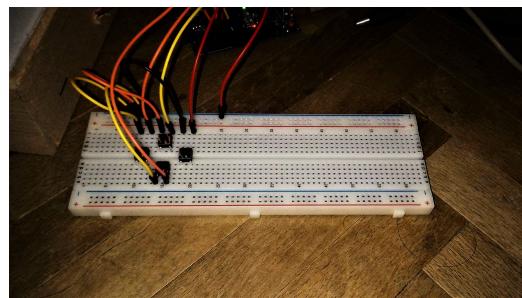


Figura 11: Controller-ul jocului Snake.

### 3 Implementare cod

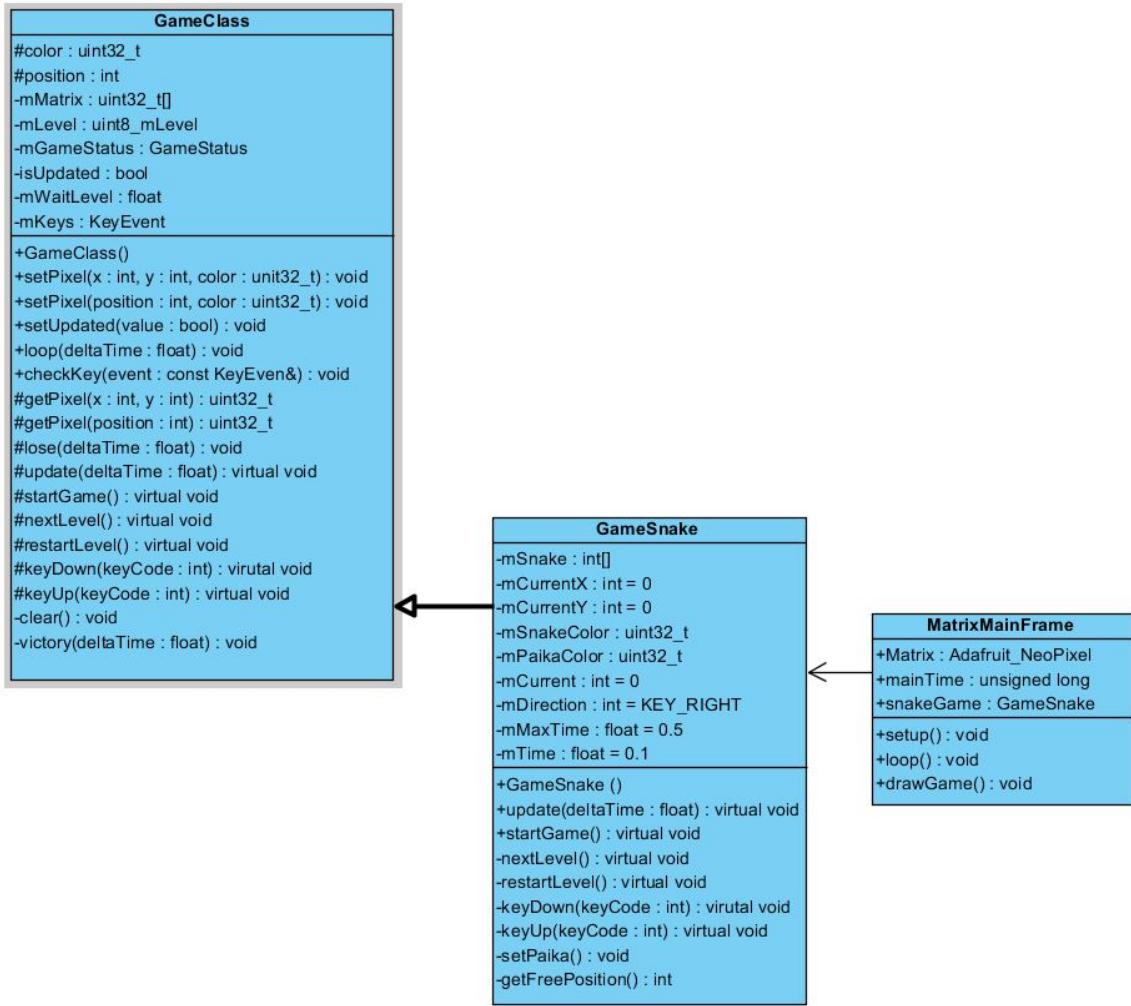


Figura 12: Diagrama de clase.

Implementarea constă în trei clase:

1. MatrixMainFrame
2. GameClass
3. GameSnake

Clasa **MatrixMainFrame** este fișierul sketch al proiectului (.INO). Se află într-o relație de asociere unidirecțională cu clasa **GameClass**, având ca instanță un atribut de tip GameClass, respectiv game. Această instanță e folosită pentru inițializarea jocului. Se ocupă de asemenea definirea codului butoanelor, colorării matricei de leduri și mișcării acesteia.

Clasa **GameClass** se ocupă de mecanica jocului, verifică mișările posibile ce duc sau nu la pierderea sau câștigarea jocului. Această supraclasă este extinsă de clasa **GameSnake**.

**GameSnake** se ocupă de managementul *snake*-ului: culoare, mișări, poziție de start, dimensiune. Dimensiunea vectorului *snake* se modifică odată cu coliziunea unui obiect de tip *Paika*, al cărui comportament este de asemenea implementat în această clasă.

Condiția de terminare a jocului apare atunci când playerul a apăsat pe un buton ce se mișcă cotrar direcției de deplasare actuale a personajului sau dacă acesta interferează cu propriul corp ceea ce duce la apariția unui ecran de *Game Over*.

Jocul se află într-un loop ce îl determină să continue la infinit.

## 4 Rezultate

În final s-a obținut un joc de Snake ce se desfășoară pe o matrice de  $8 \times 8$  unități, iar movement-ul este asigurat de controller.

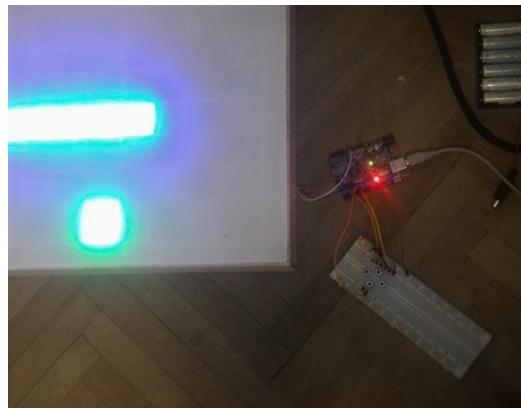


Figura 13: Snake game.

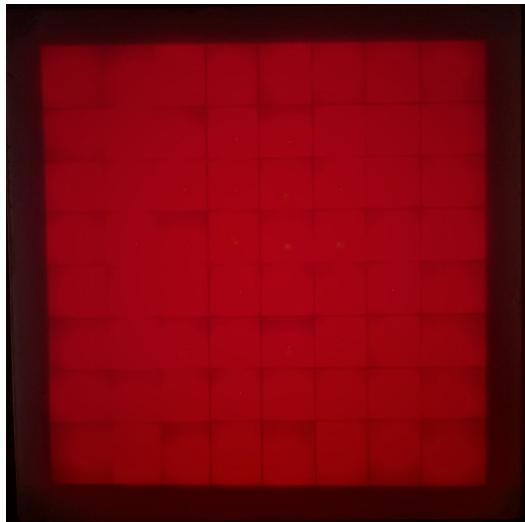


Figura 14: Game Over!

## 5 Concluzii

În concluzie, construirea și implementarea jocului a fost un mod interesant și distractiv de dobândire sau revizuire de conoștiințe în materie de Arduino și C++.