

PAC-MAN

Introducere

Proiectul „**PAC-MAN Portable**” constă în realizarea unei mini-console portabile de joc, inspirată de titlul clasic PAC-MAN, implementată pe platforma Arduino Uno R3 (ATmega328P). Sistemul îmbină mai multe componente hardware și software, oferind funcționalități integrate precum afișare grafică pe display OLED, interacțiuni prin butoane fizice, reglaj de viteză prin potențiometrul, feedback vizual prin LED RGB, și efecte sonore prin buzzer.

Acest proiect are rolul de a demonstra integrarea și utilizarea eficientă a diferitelor interfețe hardware și protocoale de comunicație serială, fiind totodată un exemplu aplicat de dezvoltare embedded cu funcționalități avansate de joc.

Funcționalități cheie

- Control precis al graficii 2D pe un display monocrom cu animații fluide
- Reglaj de viteză în timp real prin potențiometrul analogic pentru ajustarea dificultății
- Salvare / încărcare automată a progresului jocului pe card microSD
- Feedback vizual prin LED RGB, în funcție de starea jocului (victorie / eșec / imunitate)
- Efecte sonore pentru acțiuni (power-up, game over, victorie)
- 4 nivele progresive cu complexitate crescută (capcane, following ghost, power-ups)
- Sistem de imunitate temporară cu efecte vizuale speciale (LED RGB Blue pe PWM)

Descriere generală

Consola este alcătuită dintr-un modul principal **Arduino Uno R3** (ATmega328P), un **display OLED** 128×64 I2C (SSD1306), un **LED RGB** pentru feedback vizual, **5 butoane** tactile pentru controlul jocului, un **buzzer** pentru efecte sonore și un **potențiometrul** pentru ajustarea vitezei. De asemenea, include un **modul SD card** conectat prin SPI pentru salvarea automată a progresului. Toate componentele sunt montate pe **breadboard** pentru testare rapidă, apoi pe o **placă de prototipare**. Alimentarea se face fie prin **USB**, fie printr-o **baterie** de **9V** conectată la Arduino printr-un suport cu **jack** (OKY0259).



Hardware Design

Componente:

- Arduino Uno R3 (ATmega328P)
- Display OLED 128×64 I2C (SSD1306)
- LED RGB (anod comun)
- Rezistențe 220Ω
- Butoane tactil
- Rezistențe 10kΩ (pentru pull-down la butoane)
- Breadboard (plăcuță de test)
- Placă prototipare 7×9
- Cabluri Dupont m-m
- Cablu USB A-B
- Modul Card SD (SPI)
- Card microSD + adaptor SD
- Potențiomtru 10kΩ
- Rezistență 10kΩ (pentru potentiometru)
- Suport baterie 9V cu conector DC (OKY0259)
- Baterie 9V

Protocoale și Interfețe Hardware:

1. GPIO
 - Utilizat pentru citirea butoanelor și controlul LED-urilor
2. Timere / PWM
 - Utilizat pentru controlul intensității culorilor LED-ului RGB prin generarea de semnal digital cu factor de umplere variabil
3. ADC
 - Utilizat pentru a citi valorile analogice de pe potențiomtru (valori între 0-1023, rezultate din conversia tensiunii analogice 0-5V în valori digitale pe 10 biți)
4. SPI
 - Utilizat pentru comunicarea cu modulul microSD; interfață rapidă, full-duplex, pe patru linii principale (MOSI, MISO, SCK, CS)
5. I2C
 - Utilizat pentru comunicarea cu display-ul OLED; două fire (SDA, SCL) partajate între dispozitivele master / slave

Bill of Materials (BOM)

Componentă	Tip	Protocol	Cantitate	Link / Specificații
Arduino UNO R3	Controller	-	1	datasheet
Buton (Sus - D2)	Input	GPIO	1	link-piesa
Buton (Jos - D3)	Input	GPIO	1	link-piesa
Buton (Stânga - D4)	Input	GPIO	1	link-piesa
Buton (Dreapta - D5)	Input	GPIO	1	link-piesa
Buton (Shoot - D6)	Input	GPIO	1	link-piesa
OLED Display 128 x 64	Output	I2C	1	link-piesa
RGB LED (anod comun)	Output	GPIO / PWM	1	link-piesa

Potențiomtru (10kΩ)	Input	ADC	1	link-piesa
Buzzer (pe A1)	Output	GPIO	1	link-piesa
Card SD + Modul	I/O	SPI	1	link-piesa
Breadboard 170 puncte	-	-	1	link-piesa
Placă prototipare 7×9	-	-	1	link-piesa
Suport baterie 9V	-	-	1	link-piesa
Rezistență 220Ω - LED	-	-	3	-
Rezistență 10kΩ - butoane	-	-	5	-

Schemă Electrică



Descriere Detaliată

- Cele 5 butoane de control (sus, jos, stânga, dreapta, start / restart) sunt conectate pe pinii digitali D2-D6 cu rezistențe pull-up interne și sunt verificate constant pentru a controla jucătorul prin labirint
- Potențiometrul este conectat pe pinul analogic A0 și este folosit pentru a regla viteza jocului (dificultatea), fiind citit prin ADC și mapată între valorile minime și maxime de delay
- Potențiometrul este folosit de asemenea și ca seed pentru generarea punctelor aleator pe mapă
- Display-ul OLED este conectat prin I2C folosind SDA (A4) și SCL (A5) pentru afișarea labirintului, jucătorului, punctelor, fantasmei și efectelor speciale
- LED-ul RGB este conectat la D7 (roșu), D8 (verde), D9 (albastru), unde D7 și D8 sunt digitali, iar D9 este PWM pentru efecte de fading (game over roșu + pwm manual, victorie verde + pwm manual, imunitate albastru + pwm hardware)
- Buzzer-ul este conectat la A1 și folosit pentru generarea de sunete cu frecvențe diferite (power-up, game over, victory)
- Cardul SD este conectat prin SPI folosind pinii standard: MISO (D12), MOSI (D11), SCK (D13) și CS (D10) pentru salvarea automată și încărcarea progresului jocului la restart

Caracteristici Advanced ale Jocului

- **Nivelul 1:** Labirint basic cu 9 puncte de colectat
- **Nivelul 2:** Introducerea capcanelor
- **Nivelul 3:** Adăugarea unei fantome simple care urmărește jucătorul
- **Nivelul 4:** Nivel complet cu fantoma, power-up special pentru imunitate temporară de 5 secunde (protecție împotriva fantomei)
- **Sistem de Salvare:** Progresul se salvează automat la fiecare mișcare, permițând continuarea jocului după restart
- **Efecte Vizuale:** Jucătorul pulsează când are imunitate, LED-ul albastru face efecte de fading
- **Audio Feedback:** Sunete distincte pentru fiecare acțiune (power-up, game over, win, dot collection)

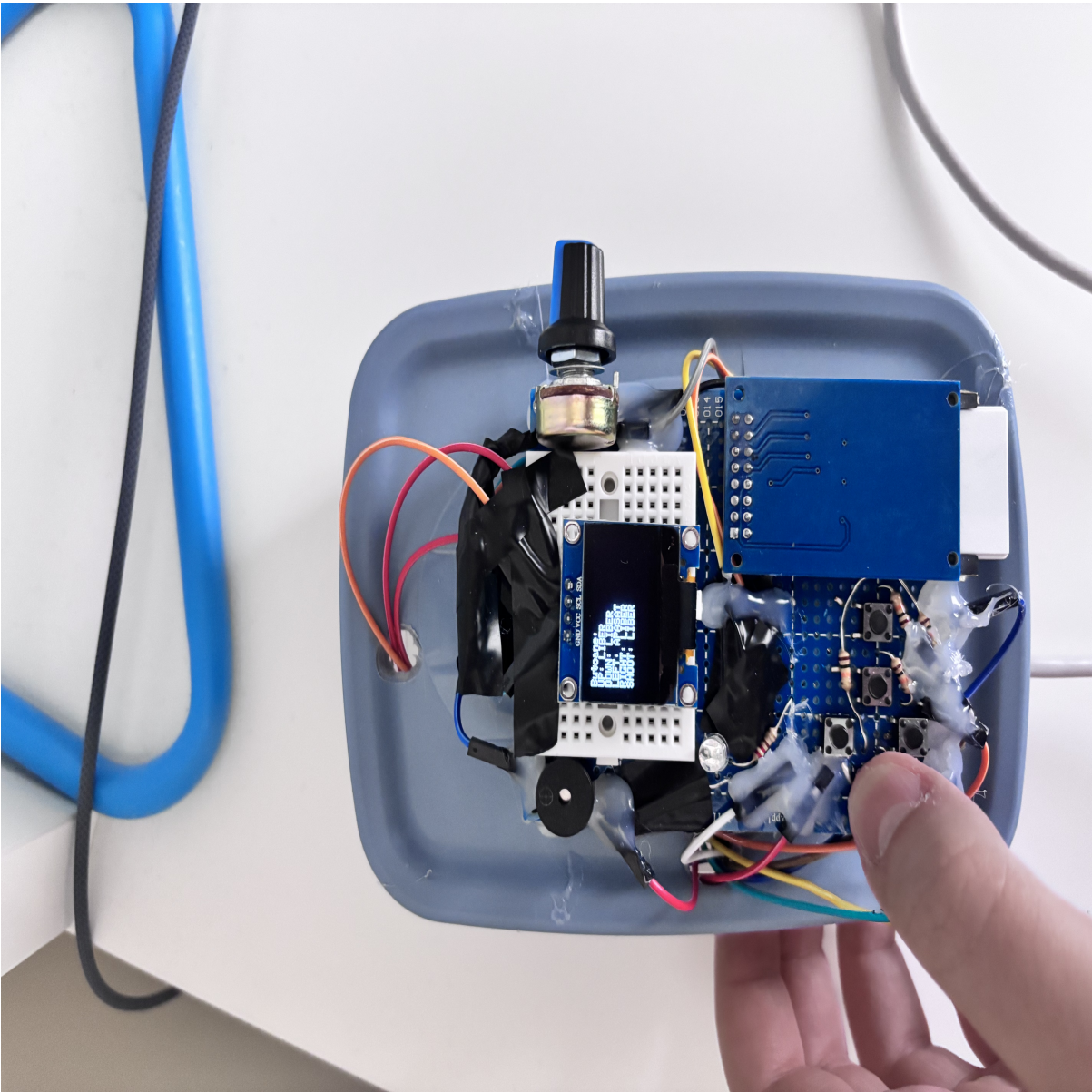
Tabel Conexiuni Componente

Componentă	Pin Arduino	Funcție
Buton Sus	D2	Control joc - mișcare sus (digital input)
Buton Jos	D3	Control joc - mișcare jos (digital input)
Buton Stânga	D4	Control joc - mișcare stânga (digital input)
Buton Dreapta	D5	Control joc - mișcare dreapta (digital input)

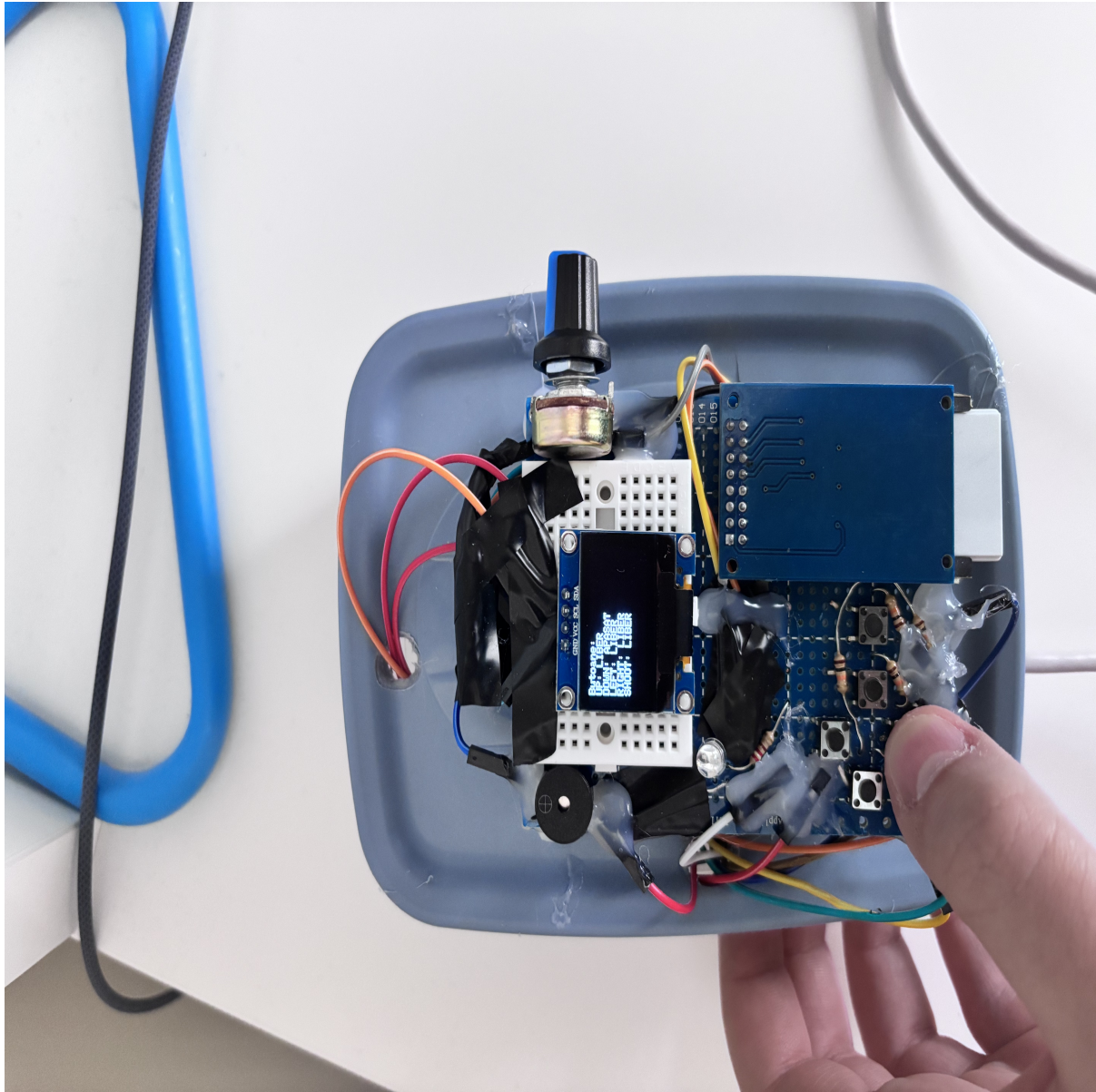
Buton Shoot	D6	Declanșare împușcare (digital input)
Potențiomtru	A0	Reglare dificultate - citire analogică (ADC)
Display OLED	A4 (SDA), A5 (SCL)	Afișaj stare joc
LED RGB - Roșu	D7	Feedback - stare eșec (digital output)
LED RGB - Verde	D8	Feedback - stare victorie (digital output)
LED RGB - Albastru	D9 (PWM)	Feedback - efecte fading (PWM output)
Buzzer	A1	Sunete joc (digital output)
SD Card - MISO	D12	Citire date (SPI)
SD Card - MOSI	D11	Scriere date (SPI)
SD Card - SCK	D13	Ceas SPI
SD Card - CS	D10	Chip Select pentru SD (SPI)

Imagini Sugestive Implementare Hardware

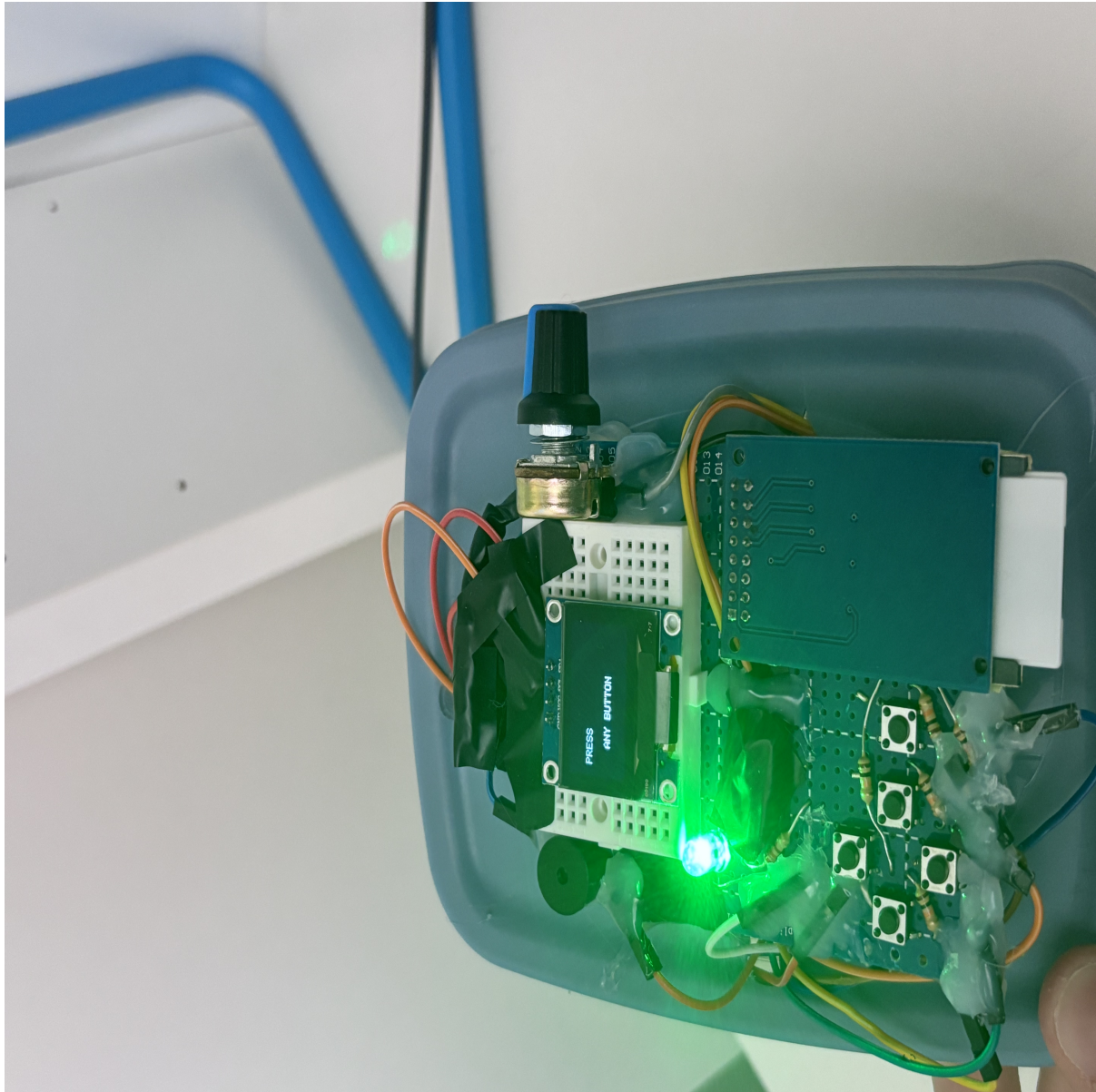
- Test Display & Butoane



- Test Display & Butoane



- Test Display & LED - verde



Software Design

Mediu de dezvoltare

- **Arduino IDE**
- Platformă de dezvoltare compatibilă cu Arduino UNO (ATmega328P), utilizată pentru scrierea, compilarea și încărcarea codului în microcontroler.

Librării și surse 3rd-party

- **Adafruit_GFX.h** - librărie grafică utilizată pentru desenarea formelor și simbolurilor pe ecrane OLED.
- **Adafruit_SSD1306.h** - driver pentru afișajele OLED cu controller SSD1306 (comunicare I2C).

- **Wire.h** – librărie standard pentru comunicarea I2C.
- Toate librăriile sunt disponibile prin Library Manager în Arduino IDE sau deja incluse implicit în cazul SPI/Wire.

Componente hardware utilizate

- **Display OLED 128×64** (conectat prin I2C pe pinii A4 – SDA, A5 – SCL)
- **5 Butoane** pentru control direcțional și acțiune (pini D2-D6)
- **Potentiometru** pentru controlul vitezei jocului (A0)
- **LED RGB** pentru feedback vizual (pini D7 – roșu, D8 – verde, D9 – albastru, PWM)
- **Buzzer** pentru semnale audio (pin A1)
- **Modul SD Card** conectat prin SPI pe pinii MISO - D12, MOSI - D11, SCK - D13, CS - D10

Algoritmi și structuri implementate

- **Harta jocului** este definită sub formă de matrice 2D (wallsMap), cu dimensiuni 16×8.
- **Generarea punctelor** - La începutul fiecărui nivel, se generează puncte de colectat (DOT-uri) aleator pe poziții libere.
- **Reprezentarea jucătorului** - Jucătorul este reprezentat de un cerc desenat pe ecran, care se deplasează în funcție de inputul de la butoane.
- **Sistemul fantomei** - O fantomă apare în nivelurile 3 și 4, deplasându-se pentru a urmări jucătorul.
- **Power-up și imunitate** - La nivelul 4, este introdus un tile special de tip power-up, care activează imunitatea temporară (5 secunde) în care fantoma este de asemenea înghețată.

Efecte imunitate:

- Fantoma nu mai produce Game Over la coliziune
- Fantoma este înghețată
- LED-ul albastru pulsează
- Jucătorul are o animație grafică specială

Condiții de terminare:

- **Victorie:** Toate DOT-urile au fost colectate (LED verde + sunet de victorie)
- **Game Over:** Jucătorul este prins de fantomă sau calcă într-o capcană (LED roșu + sunet specific)

Funcții și comportamente principale

setup()

- Inițializează comunicațiile I2C și seriale
- Inițializează display-ul OLED și LED-urile
- Configurează pinii pentru butoane și buzzer
- Afișează ecranul de Start

loop()

- Monitorizează inputul de la butoane pentru mișcarea jucătorului
- Actualizează poziția fantomei (dacă este activă)
- Verifică coliziunile: cu pereți, puncte, capcane, fantome
- Desenează constant harta actualizată și toate elementele vizuale
- Activează și dezactivează automat imunitatea în funcție de timp
- Controlează LED-urile în funcție de starea jocului
- Generează efecte audio prin buzzer pentru evenimentele importante

generateDots(uint8_t count)

- Populează harta cu un număr de puncte colectabile
- Adaugă capcane și fantoma în nivelurile superioare
- În nivelul 4, adaugă un tile power-up într-o poziție fixă

drawTile(), drawPlayer(), drawGhost()

- Redau grafic elementele jocului pe OLED: pereți, puncte, capcane, power-up, jucător, fantomă
- Jucătorul primește o animație specială în timpul imunității

isImmunityActive()

- Returnează true dacă jucătorul este încă în perioada de imunitate
- Dezactivează automat imunitatea după 5 secunde și oprește LED-ul albastru

playPowerUpSound(), playGameOverSound(), playWinSound(), playDotSound()

- Emit tonuri sonore diferite prin buzzer în funcție de evenimentul declanșat

Funcționalități speciale

Controlul vitezei jocului printr-un potentiometru (analog A0), adaptând delay-ul de mișcare.

Feedback vizual și audio complet:

- LED verde: câștig
- LED roșu: pierdere
- LED albastru: imunitate
- Buzzer: notificări sonore

Extindere modulară pentru niveluri multiple (până la 4), fiecare cu logică specifică.

Descrierea codului aplicației (firmware):

- **Mediu de dezvoltare:** Arduino IDE
- **Librării și surse 3rd-party:** Adafruit_GFX, Adafruit_SSD1306, SPI, Wire
- **Algoritmi și structuri implementate:** Matrice 2D pentru hartă, algoritm de urmărire pentru fantomă, sistem de imunitate temporară, generare aleatoare de puncte
- **Surse și funcții implementate:** Gestionare completă a jocului cu 4 nivele progresive, sistem de feedback vizual / audio, control prin butoane și potențiometru

Rezultate Obținute

În urma realizării proiectului am reușit următoarele milestone-uri:

- **Hardware:**
 - Butoane conectate corect și funcționale
 - Display OLED conectat corespunzător și funcțional, atât că am conectat pe un breadboard, ulterior, din cauza unor mici probleme cu placa de prototipare
 - Buzzer funcțional
 - Potențiometru funcțional
 - LED RGB anod comun funcțional și cu PWM pe albastru
 - Modul Card SD conectat corect
- **Software:**
 - 4 niveluri progresive din punct de vedere al dificultății
 - Semnale vizuale și sonore prin LED, respectiv buzzer, pentru câștig, pierdere și activarea funcției de imunitate
 - Viteză ajustabilă prin potențiometru în funcție de nivel
 - Efect vizual pe OLED al jucătorului cât timp se află sub protecția imunității
 - Modulul SD Card nu a fost implementat

Concluzie

În concluzie, am încercat să fac acest joc cât mai interactiv, de aceea am folosit și funcția de imunitate la nivelul 4 astfel încât jucătorul să re trăiască adrenalina jocului clasic pac-man. Am adăugat diverse efecte vizuale / audio astfel încât jocul să îi țină pe cei care îl încearcă cât se poate de captivați și dornici să îl termine. Totodată nu am optat pentru foarte multe niveluri, astfel încât să

nu intervină monotonia, ci am mers pe 4 niveluri, care cresc gradual ca dificultate, iar finalul reprezintă un apogeu după suspansul avut cât timp jucătorul se află sub efectul imunității. Acesta este proiectul implementat, denumit **"PAC-MAN Portable"**.

Bibliografie/Resurse

Resurse Hardware

- [ATmega328P Datasheet](#)
- [Placa Arduino UNO R3](#)
- [Modul microSD](#)
- [Card microSD](#)
- [Display OLED](#)

Resurse Software

- [Arduino IDE](#)
- [Librărie SPI.h](#)
- [Documentație Arduino UNO](#)

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/apredescu/florian.ariasu>



Last update: **2025/05/28 09:25**