

Run, Billy, run

Proiect realizat de Adina-Maria Amzarescu, 331CA

Introducere

Prezentarea proiectului:

- Acest proiect reprezinta un joc 2D din categoria jocurilor “endless runner”
- Scopul jucatorului este de a depasi obstacolele in miscare si a acumula puncte
- Inspiratia vine din jocurile de tip “endless runner” [2] si are un sistem asemanator al jocului [Dinosaur Game](#)
- Jocurile de acest tip sunt facute pentru a tine utilizatorul atent si reprezinta o sursa imediata de dopamina

Motivatia pentru realizarea acestui proiect:

Acest proiect isi doreste sa documenteze atat modul de realizare al unui joc de tipul endless runner, cat si interactiunea utilizatorului cu acest tip de jocuri.

Jocurile de tip “endless runner” ofera un cadru ideal pentru a pune jucatorul intr-o stare de “Flow” [1]. Alergarea utilizatorului creeaza rezultate rapide, iar astfel acesta se simte “skilled”, rezultand intr-o oarecare dependenta de acest tip de jocuri.

Laboratoare utilizate:

1. USART. Digital Debugging
2. Analog Digital Convertor (ADC) pentru leduri
3. Întreruperi hardware. Întreruperi externe pentru joystick
4. I2C (Inter-Integrated Circuit) pentru LCD

Descriere generală

Schema bloc a proiectului:

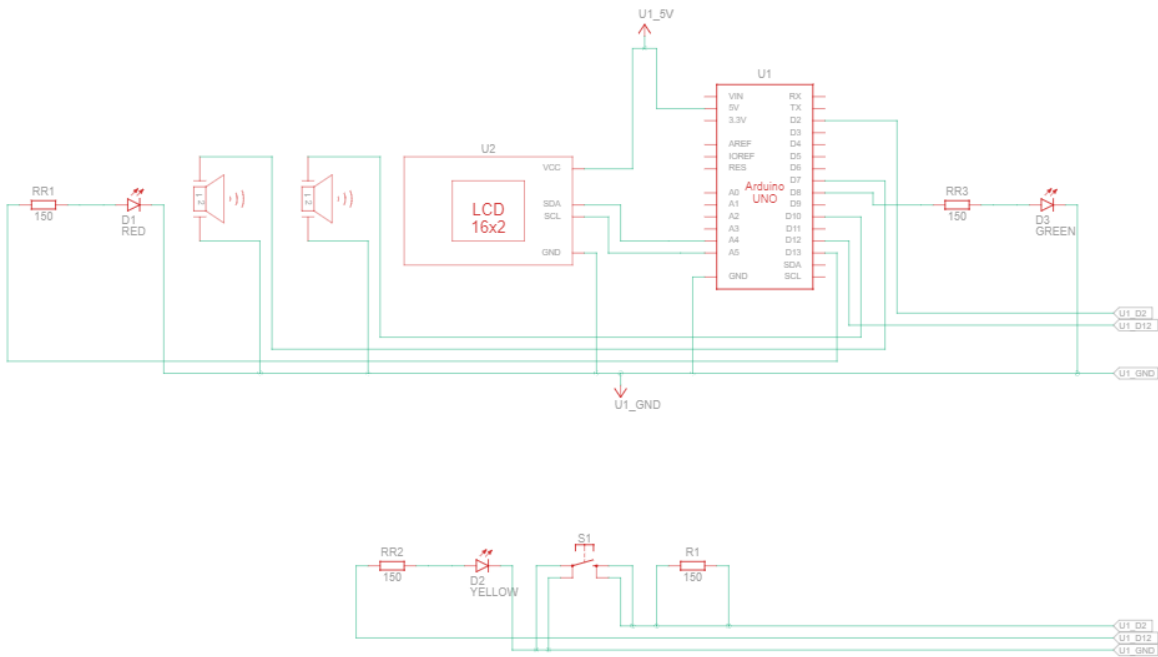


Jocul este o varianta a unui joc de tipul “endless runner”. Jocul se va desfasura pe un LCD 16x2, utilizatorul va controla miscarea playerului dintr-un joystick. Proiectul foloseste o placuta Arduino UNO R3. Proiectul va avea de asemenea si 3 LED-uri (rosu, galben, verde) pentru a-i indica utilizatorului

cand va incepe jocul si pentru a-i indica daca a depasit sau nu obstacolul. De asemenea va fi un buzzer pentru a semnaliza sonor depasirea unui obstacol.

Hardware Design

Schema electrica:



Componentele principale:

- Arduino UNO R3



- LCD 16x2 cu I2C



- Joystick compatibil cu arduino



- 3 LED-uri



Alte componente:

- Buzzer activ OKY 0151



- Breadboard



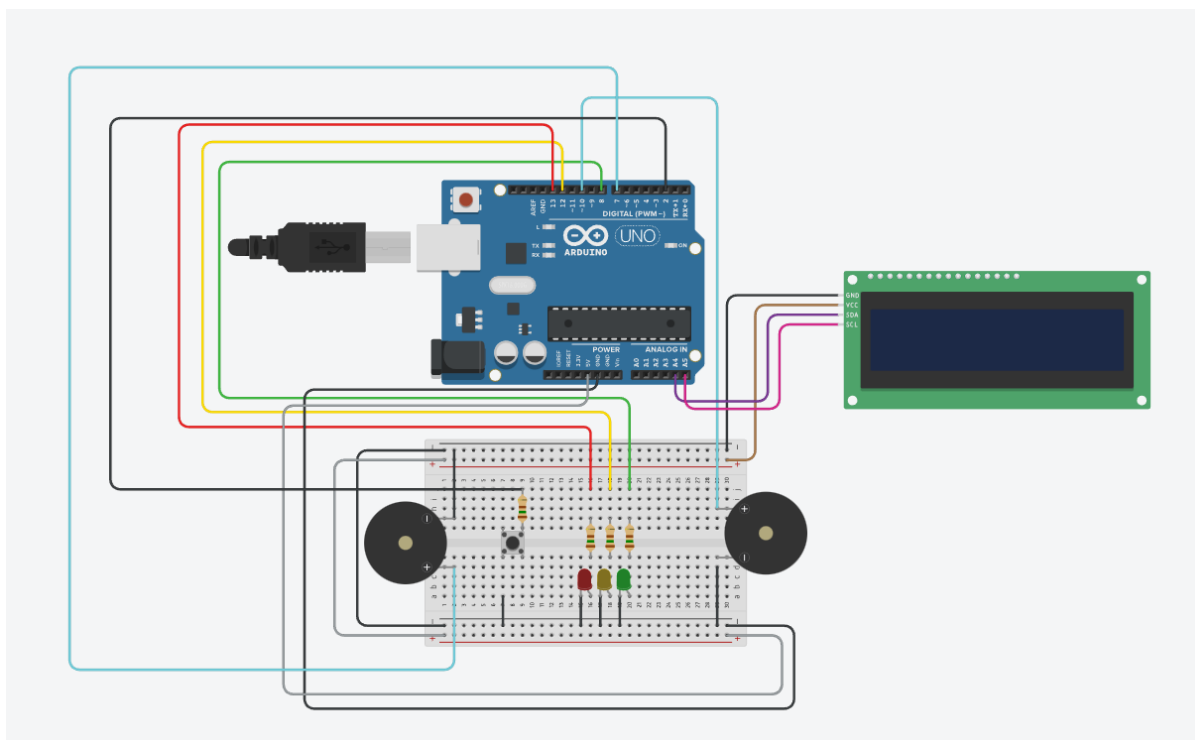
- Fire conexiune



- Rezistente



Proiectul realizat in TinkerCad:



[Link catre simulare.](#)

Mentionez faptul ca in TinkerCad nu exista un modul pentru Joystick, motiv pentru care am folosit un buton pentru a putea simula.

Montarea pe placuta:

LCD I2C:

- GND → GND
- VCC → 5V
- SDA → A4
- SCL → A5

Modulul Joystick:

- GND → GND
- 5V → 5V
- VRx → D2

Alert Buzzer:

- - → GND
- + → D10

Music Buzzer:

- - → GND
- + → D7

LED-uri:

- Rosu: D13
- Galben: D12
- Verde: D8
- GND → GND

Software Design

Realizarea proiectului:

Proiectul va fi dezvoltat in limbajul C++ in **Arduino IDE 2.1.0**.

Librarii auxiliare:

- * New-LiquidCrystal
- * ProtoStax_MmlMusicPlayer

Link github pentru librariile folosite:

[New-LiquidCrystal](#)

[ProtoStax_MmlMusicPlayer](#)

Modul de joc:

Ideea codului este de a deplasa terenul la stanga, nu jucatorul la dreapta. Jucatorul va sta pe loc si se va deplasa doar sus si jos, folosind joystick-ul.

Inainte sa inceapa jocul, pe ecran va fi afisat textul: *Push to Start*. Cand utilizatorul va apasa pe joystick terenul se va genera si deplasa la stanga. Inainte de generarea terenului, cele 3 leduri se vor aprinde pe rand, ultimul, cel verde, semnaland inceperea jocului.

Aprinderea ledurilor a fost implementata folosind Analog Digital Convertor (ADC).

```
// RED
lcd.setCursor(0,0);
lcd.print("Start in 3  ");
for (int intensity = 0; intensity <= 255; intensity++) {
  analogWrite(RED_PIN, intensity);
  delay(5);
}
for (int intensity = 255; intensity >= 0; intensity--) {
  analogWrite(RED_PIN, intensity);
  delay(5);
}
Serial.print("RED LED DONE");
// YELLOW
lcd.setCursor(0,0);
lcd.print("Start in 2  ");
for (int intensity = 0; intensity <= 255; intensity++) {
  analogWrite(YELLOW_PIN, intensity);
  delay(5);
}
for (int intensity = 255; intensity >= 0; intensity--) {
  analogWrite(YELLOW_PIN, intensity);
  delay(5);
}
Serial.print("YELLOW LED DONE");
// GREEN
lcd.setCursor(0,0);
lcd.print("Start in 1  ");
for (int intensity = 0; intensity <= 255; intensity++) {
  analogWrite(GREEN_PIN, intensity);
  delay(5);
}
for (int intensity = 255; intensity >= 0; intensity--) {
  analogWrite(GREEN_PIN, intensity);
  delay(5);
}
delay(5);
Serial.print("GREEN LED DONE");
```

Pentru a depasi obstacolele, utilizatorul trebuie sa mute pe axa Ox joystick-ul. Pentru a detecta apasarea butonului am folosit o intrerupere, urmata de functia buttonPush.

```
// Attach an interrupt to interrupt 0 (Digital pin 2) to detect button push
attachInterrupt(0, buttonPush, FALLING);
```

Punctele obtinute sunt reprezentate de numarul de secunde petrecute in joc si vor fi afisate in partea dreapta a ecranului.

De fiecare data cand player-ul sare actiunea sa va fi anuntata de un semnal sonor.

In momentul in care player-ul se loveste de un obstacol, jocul se opreste.

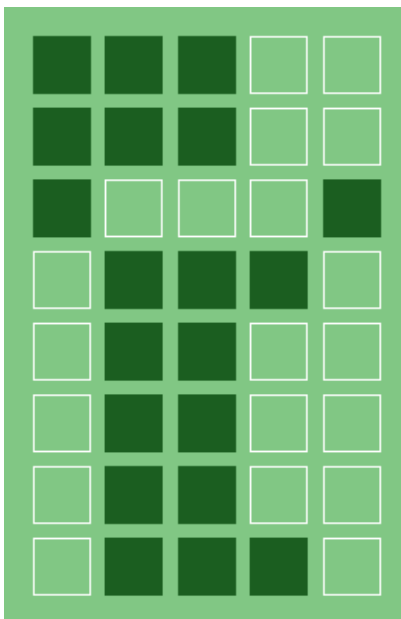
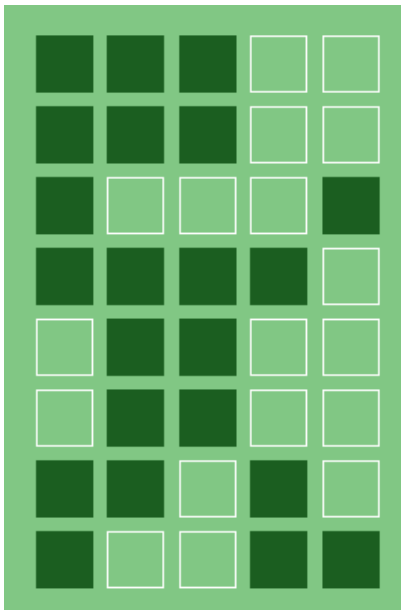
In timpul jocului va fi muzica pe fundal. Pentru muzica am folosit [Musescore](#)

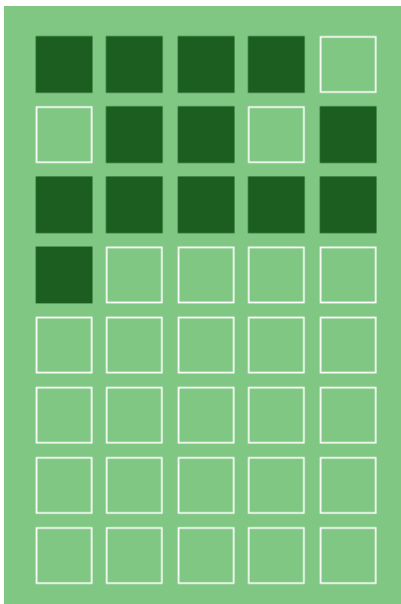
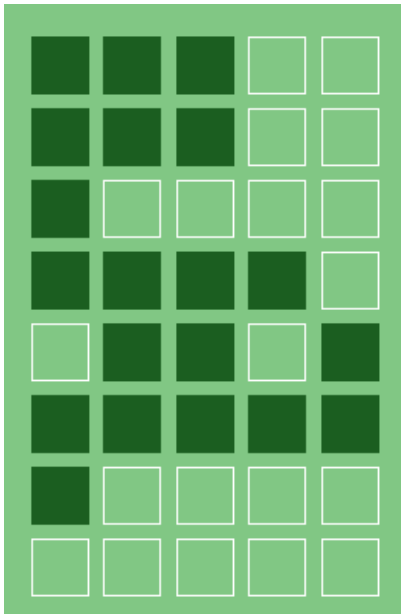
* T160 - tempo-ul

* O4 - octava 4

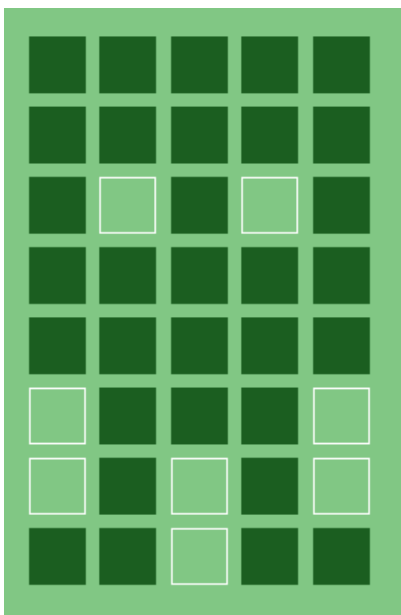
Pentru realizarea caracterului si a obstacolelor am folosit [LCD-Character-Creator](#)

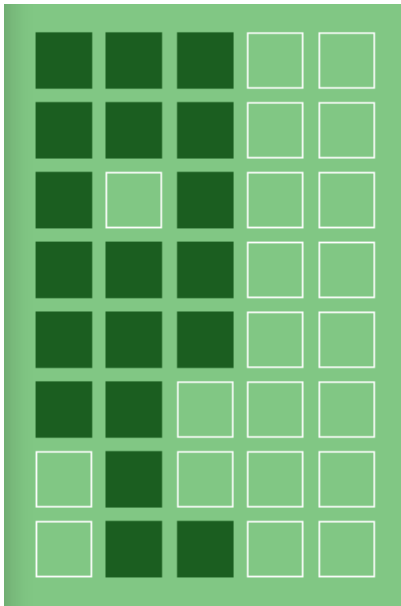
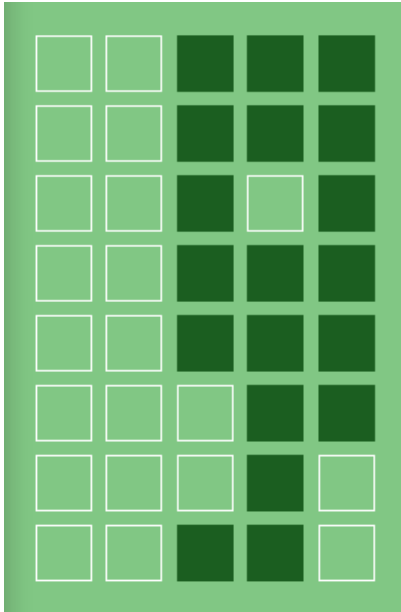
Player:





Obstacle:





Functii:

1. initializeGraphics():

Această funcție inițializează grafica personalizată pentru afișajul LCD prin crearea de modele de caractere bazate pe matrice de octeți predefinite. Setează reprezentări grafice pentru mișcările jucătorilor, terenul și copacii.

2. advanceTerrain():

Această funcție este responsabilă pentru alunecarea terenului spre stânga în trepte de jumătate de caracter. Acesta actualizează matricele de teren pe baza terenului actual și a noului teren.

3. drawPlayer():

Această funcție atrage jucătorul pe afișajul LCD în funcție de poziția jucătorului și actualizează caracterele terenului în consecință. De asemenea, verifică dacă există coliziuni între jucător și teren.

4. Bucla principală:

Bucla principală a programului actualizează continuu starea jocului, verifică intrarea pe joystick, avansează terenul, actualizează poziția jucătorului și redesenează scena pe afișajul LCD. De asemenea, se ocupă de redarea muzicii și controlul LED-ului în funcție de starea jocului.

Pentru debug am folosit:

USART. Digital Debugging

```
// Print debug information
Serial.print("Character Position: ");
Serial.println(player_position);
Serial.print("Distance: ");
Serial.println(distance);
```

Rezultate Obținute

Concluzii

1. As fi dorit sa aleg un proiect mai complex

Deși inițial credeam că o să-mi fie greu să realizez proiectul și am vrut să încep cu ceva mic, ulterior am realizat că nu era atât de greu pe cât mă așteptam și acum, dacă ar fi să reiau tot procesul, aș alege un proiect mai complex.

2. Documentarea e importantă

Atât înainte de alegerea proiectului, cât și după, ar fi trebuit să mă documentez mai mult. M-am bazat pe faptul că voi putea realiza totul de la 0 fără ajutor extern, însă când a venit vorba de muzică, am stat 2 zile să încerc să realizez muzica, iar ulterior am reușit, folosindu-mă de o bibliotecă auxiliară.

3. Piesele trebuiau comandate mai devreme

Am avut probleme cu piesele deoarece au ajuns mai tarziu decat m-am asteptat.

4. Organizarea

Am modificat proiectul pe parcurs, adaugand mai multe piese, initial voiam sa fie doar ecranul si joystick-ul, insa ulterior am vrut sa il fac mai complex, am adaugat lumini si sunete. Daca ma organizam mai bine de la inceput as fi stiut din timp de ce piese am nevoie si cum vreau sa arate proiectul.

5. Nu doresc sa ramana asa

Imi place mult jocul, chiar daca este simplu, doresc sa il imbunatatesc ulterior si sa adaug mai multe posibilitati de joc. As vrea sa mai adaug un ecran LCD care sa contina scorul separat, sa fie un meniu de joc, cu mai multe jocuri posibile. As vrea sa inlocuiesc ecranul cu unul grafic si sa modific optiunile de pe joystick astfel incat sa permita mai multe optiuni, de exemplu sa te lasi in jos, sa sari mai mult daca tii apasat mai mult timp in sus, sa existe posibilitatea modificarii vitezei pe axa OY.

6. Jocul provoaca dependenta

Ma asteptam la acest lucru, poate este si faptul ca este proiectul meu si am muncit si de aceea m-as juca incontinuu, insa imi place si nu m-as opri.

Download

Toate detaliile proiectului se regasesc in urmatoarul folder:

- [Link Drive](#)
- [codul](#)

Jurnal

- 01.05.2023 - Alegerea proiectului
- 06.05.2023 - Realizarea documentatiei initiale
- 06.05.2023 - Organizarea componentelor Hardware



Mentionez faptul ca modulul buzzer active OKY 0151 a ajuns ulterior.

- 15.05.2023 - Realizarea primei etape pentru componentele hardware



- 16.05.2023 - Realizarea etapei 2 pentru componentele hardware



- 17.05.2023 - Realizarea etapei 3 pentru componentele hardware



- 17.05.2023 - Realizarea unui prim concept de grafica



- 19.05.2023 - Realizarea partii software
- 21.05.2023 - Inlocuirea celor 3 leduri cu un modul semafor
- 22.05.2023 - Includerea muzicii + inca un buzzer

Bibliografie/Resurse

Motivatia proiectului:

[1] [https://en.wikipedia.org/wiki/Flow_\(psychology\)](https://en.wikipedia.org/wiki/Flow_(psychology))

[2] https://en.wikipedia.org/wiki/Category:Endless_runner_games

Resurse Hardware:

[3] <https://www.robofun.ro/>

[4] <https://hobbymarket.ro/>

[5] <https://cleste.ro/>

Tutoriale:

[6] <https://www.youtube.com/watch?v=j0n7DIpSv4>

Resurse Software:

[7] <https://www.arduino.cc/>

[8] <https://github.com/fmalpartida/New-LiquidCrystal>

[9] <https://maxpromer.github.io/LCD-Character-Creator/>

[10] <https://musescore.com/>

Tutoriale:

[11] https://github.com/protostax/ProtoStax_MmlMusicPlayer

[12] <https://www.youtube.com/watch?v=NbXp7luU9NU&t=78s>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/runbillyrun>



Last update: **2023/05/30 10:33**