

Chrome Dinosaur Game

Autor

[Pirvulet Mihai-Dumitru](#)

Introducere

- Proiectul este inspirat din jocul deja extrem de popular care apare in Google Chrome atunci cand nu mai ai conexiune la internet. Este un endless runner cu un dinozaur care trebuie sa sara peste obstacole si sa stranga puncte.
- Scopul proiectului este realizarea unui endless runner similar, cu diverse obstacole, scor si controlat prin intermediul unor butoane, jucat direct pe Arduino.



Descriere generală

Se conectează la un Arduino UNO: un ecran LCD, un buton, un led RGB și un buzzer. Jocul este similar celui din browser atunci când îți pica netul. Ai un personaj pe care îl controlezi (l-am botezat "Jimmy", idk why) și trebuie să sara peste obstacole generate random. La fiecare obstacol sarit cu succes primești 1 punct.

LED-ul RGB indică dificultatea: verde - ușor, galben - mediu, roșu - greu, și se schimbă la pragul de 5, respectiv 10 puncte, ca să mențină jocul interesant. Buzzerul e folosit pentru feedback atunci când jucătorul sare sau primește 1 punct, precum și un cântec de încurajare (sau troll) la final de joc. Potentiometrul e folosit la setarea contrastului ecranului LCD.

Schema Bloc



Hardware Design

Listă piese

- Arduino UNO
- Ecran LCD
- Buton
- Buzzer
- LED RGB
- Potentiometru 10k
- Rezistente
- Fire mama-tata, tata-mama, tata-tata

Schema Electrica

Versiunea initiala - Tinkercad



Link Tinkercad pentru testare: <https://www.tinkercad.com/things/e6jOIGIUE3u>. Momentan este incarcat un cod de test al componentelor pentru a verifica faptul ca acestea sunt conectate corect. Logica de joc va fi implementata cand se va face trecerea in hardware. Posibile modificari ulterioare: mai multe butoane conectate similar si un led RGB.

Versiunea finala - EAGLE



Dupa implementarea efectiva a proiectului, am adus cateva modificari. Schema EAGLE reprezinta varianta finala, cu led RGB si pinii corecti.

Software Design

Medii de dezvoltare folosite

- **Arduino IDE** - pentru programarea placutei Arduino
- **Draw.io** - pentru schema bloc
- **EAGLE & TinkerCAD** - pentru schemele electrice

Biblioteci

- **LiquidCrystal.h** - pentru interactiunea cu ecranul LCD
- **pitches.h** - pentru a reda cantece pe buzzer

Descrierea Codului

Initializari - inainte de orice functie, sunt cateva variabile care controleaza flow-ul si functionalitatea jocului. Cateva din acestea sunt:

- **tempo** - controleaza viteza cu care se reda melodia la final
- **melody** - notele melodiei folosind **pitches.h**
- **jimmy, jumpingjimmy, spike, fence** - sprite-uri pentru grafica LCD-ului
- **chance** - sansa ca la un tick de joc sa se genereze un obstacol
- **gameOver** - variabila de stare care retine daca jocul s-a terminat sau nu
- **tick** - cate milisecunde dureaza un tick de joc (viteza jocului)
- **score** - initializat cu 0, in functie de acesta va creste dificultatea
- pinii pentru fiecare componenta pentru a fi folositi ulterior in cod

Funcții

- **setup()** - se initializeaza LCD-ul, sprite-urile, se seteaza dificultatea initiala pe easy (culoarea LED-ului verde) si se seteaza seedul pentru randomizarea jocului
- **loop()** - logica principala de joc, se verifica daca jocul s-a terminat, caz in care se afiseaza un ecran de game over. Altfel, se randeaza jocul, starea lui fiind retinuta in matricea **ecran[][]**, unde 0 = clear, -1 = jucator, orice altceva diferit de 0 obstacole, care sunt de 2 tipuri. De asemenea, se verifica apasarea butonului si executia logicii de sarit.
- **setColorRGB()** - functie ajutatoare pentru a seta valori pe ledul RGB din intervalul 0-255 pentru fiecare channel.
- **renderGame()** - randeaza starea jocului pe LCD
- **permutare()** - genereaza un nou frame al jocului. Mai intai da un random sa vada daca va genera un obstacol sau nu. Am adaugat un check sa nu se genereze prea multe obstacole la rand si jocul sa

fie imposibil. Dupa asta, se actualizeaza starea jocului mutand toate elementele cu o pozitie mai la stanga (permutare la stanga, de unde si numele functiei).

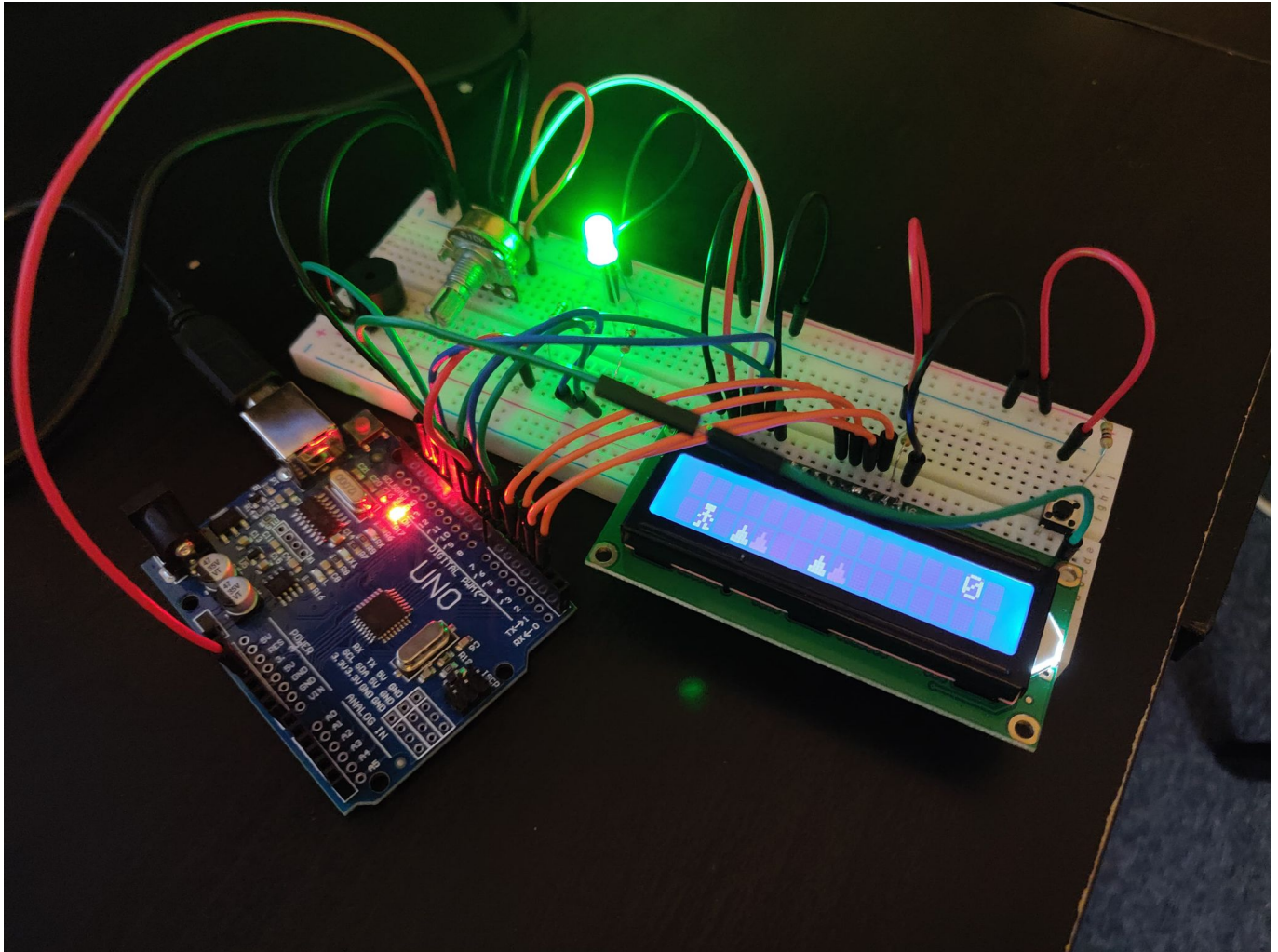
- **lose()** - afiseaza ecranul de game over, seteaza culoarea LED-ului pe rosu si reda melodia de game over.
- **updateScore()** - actualizeaza scorul jucatorului daca a sarit cu succes peste un obstacol. De asemenea, creste dificultatea la pragurile de scor 5 si 10, schimband culoarea LED-ului in galben, respectiv rosu, alaturi de cresterea vitezei.
- **jump()** - logica de sarit, cu animatie pe LCD.
- **isObstacle()** - wrapper peste functia de random care returneaza 1 in functie de sansa setata daca trebuie generat un obstacol nou
- **generate()** - returneaza un tip de obstacol random din cele setate (am apucat sa fac doar 2, dar e usor sa se adauge unele noi).
- **playSong()** - functie ce reda o melodie definita folosind **pitches.h**

Observatii

- Mi s-a parut destul de greu sa fac jocul "playable", m-am jucat foarte mult cu parametri lui pana sa ajunga intr-o versiune decenta, sa nu fie nici prea plictisitor nici imposibil.
- Un aspect de implementare interesant mi s-a parut animatia de sarit a jucatorului, care a fost destul de dificila.

Rezultate Obținute

Demo:<https://youtu.be/Rp5U3N3E26o>



Concluzii

Mie imi place ce a iesit, am stat o gramada sa configurez jocul sa fie cat de cat enjoyable (si usor addictive). Cred ca ar mai putea fi imbunatatita viteza de raspuns la inputul utilizatorului, dar overall mi s-a parut un proiect dragut care a rezumat bine ce am invatat la PM semestrul acesta.

Download

[proiect_chrome_dinosaur.zip](#)

Jurnal

- 24 Aprilie - Alegere Tema Proiect
- 27 Aprilie - Discutie cu laborantul + schimbare de la shield la componente individuale
- 24 Mai - Schema Electrica varianta initiala

- 2 Iunie - Schema Electrica varianta finala
- 2 Iunie - Finalizare pagina wiki, upload cod, poze, demo, bibliografie

Bibliografie/Resurse

- <https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody>
- Laboratoarele de PM

Pdf: https://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/chromedinosaur?do=export_pdf

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/chromedinosaur>



Last update: **2021/06/02 13:02**