

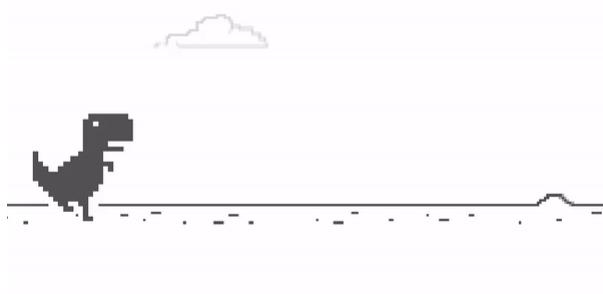
# Chrome Dinosaur Game

## 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 conecteaza la un Arduino UNO: un ecran LCD, un buton, un led RGB si un buzzer. Jocul este similar celui din browser atunci cand iti pica netul. Ai un personaj pe care il controlezi (l-am botezat "Jimmy", idk why) si trebuie sa sara peste obstacole generate random. La fiecare obstacol sarit cu succes primesti 1 punct.

LED-ul RGB indica dificultatea: verde - usor, galben - mediu, rosu - greu, si se schimba la pragul de 5, respectiv 10 puncte, ca sa mentina jocul interesant. Buzzerul e folosit pentru feedback atunci cand jucatorul sare sau primeste 1 punct, precum si un cantec de incurajare (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/e6j0IGIUE3u>. 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 ulteroare: 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 interacțiunea cu ecranul LCD
- **pitches.h** - pentru a reda cantece pe buzzer

## Descrierea Codului

**Initializari** - înainte de orice funcție, sunt câteva variabile care controlează flow-ul și funcționalitatea jocului. Câteva din acestea sunt:

- **tempo** - controlează viteza cu care se redă 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 să se genereze un obstacol
- **gameOver** - variabila de stare care retine dacă jocul s-a terminat sau nu
- **tick** - cate milisecunde durează un tick de joc (viteza jocului)
- **score** - initializat cu 0, în funcție de acesta va crește dificultatea
- pinii pentru fiecare componentă pentru a fi folosiți ulterior în cod

## Functii

- **setup()** - se initializează LCD-ul, sprite-urile, se setează dificultatea initială pe easy (culoarea LED-ului verde) și se setează seedul pentru randomizarea jocului
- **loop()** - logica principală de joc, se verifică dacă jocul s-a terminat, caz în care se afișează un ecran de game over. Altfel, se răndează jocul, starea lui fiind reținută în matricea **ecran[][]**, unde 0 = clear, -1 = jucător, orice altceva diferit de 0 obstacole, care sunt de 2 tipuri. De asemenea, se verifică apasarea butonului și execuția logicii de sarit.
- **setColorRGB()** - funcție ajutătoare pentru a seta valori pe ledul RGB din intervalul 0-255 pentru fiecare channel.
- **renderGame()** - răndează starea jocului pe LCD
- **permute()** - generează un nou frame al jocului. Mai întâi da un random să vada dacă va genera un obstacol sau nu. Am adăugat un check să nu se genereze prea multe obstacole la rand și jocul să

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

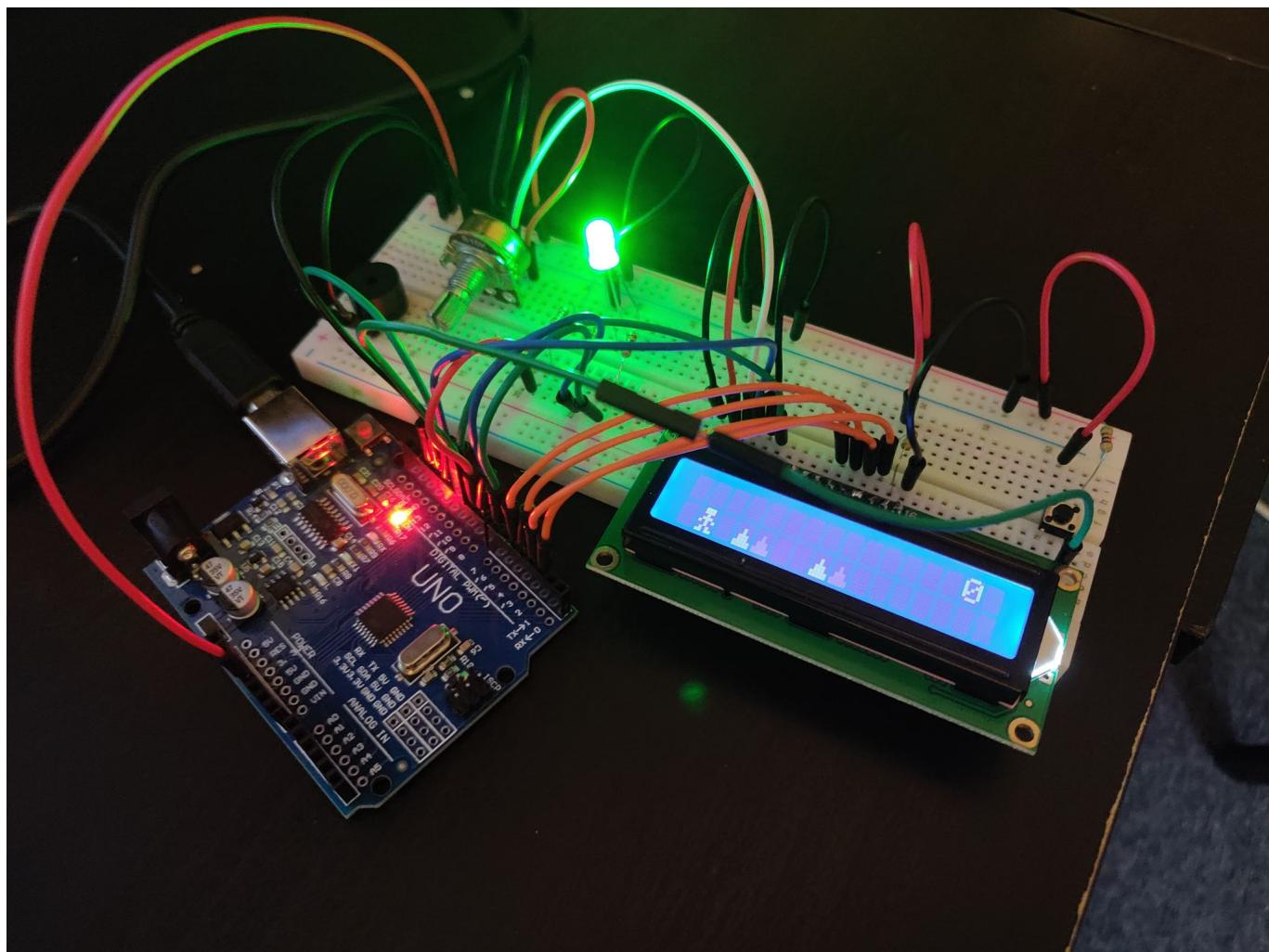
- **lose()** - afiseaza ecranul de game over, seteaza culoarea LED-ului pe rosu si reda melodia de game over.
- **updateScore()** - actualizeaza scorul jucatorului daca a sarit cu succes peste un obstacol. De asemenea, creste dificultatea la pragurile de scor 5 si 10, schimband culoarea LED-ului in galben, 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 parametrii lui pana sa ajunga intr-o versiune decenta, sa nu fie nici prea plăcătitor nici imposibil.
- Un aspect de implementare interesant mi s-a parut animatia de sarit a jucatorului, care a fost destul de dificila.

## Rezultate Obtinute

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/BuiltnExamples/toneMelody>
- Laboratoarele de PM

Pdf: [https://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/chromedinosaur?do=export\\_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**