# Dino endless runner

## Introducere

Proiectul consta intr-un joc de tip endless runner similar cu cel care apare in chrome cand cineva nu este conectat la internet. Jocul este redat pe un display, iar jucatorul are la dispozitie un buton pentru a sari ca se evite obstacolele.

Scopul proiectului este pur si simplu divertismentul. Am dorit sa recreez experienta jocurilor clasice intr-o maniera mai interesanta.

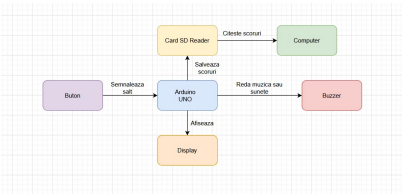
## Descriere generală

Proiectul este dezvoltat pe o placuta Arduino UNO avand ca si interfata un display mic oled.

Jucatorul controleaza actiunea jocului, adica salturile printr-un buton. Jucatorul mere in dreapta mereu, iar la anumite momente de timp, apare un obstacol peste care jucatorul trebuie sa sara. Cand jucatorul pierde, se poate restarta jocul prin apasarea butonului. De asemenea, fiecare scor este retinut pe un card SD.

Sunt redate sunete cu ajutorul unui buzzer atunci cand jucatorul sare si o melodie retro pe fundal.

Schema bloc:



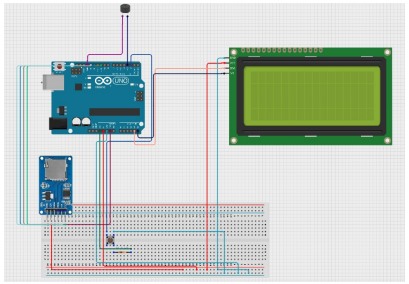

## Hardware Design

Lista de componente şi scopul lor:

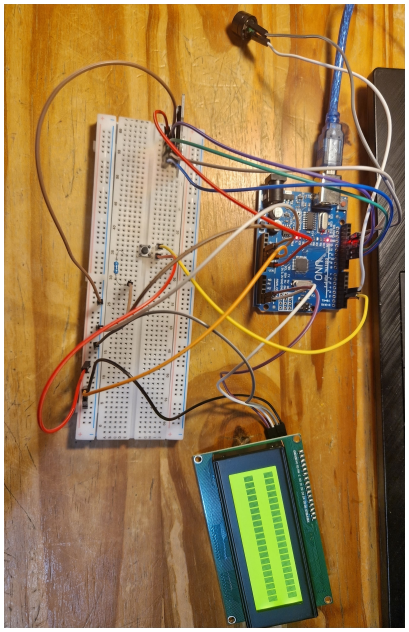
- Placa Arduino UNO → unitatea de control de acces
- Breadboard → pentru a ajuta prorotiparea şi a lega anumite componente
- LCD 4×20 → pentru a afişa jocul
- Cititor de card SD → pentru a salva toate scorurile înregistrate
- Buton → pentru a permite jucătorului să sară
- Buzzer → pentru a putea reda sunete când se întamplă ceva în joc

- Rezistori → pentru a conecta butonul
- Cabluri → pentru a lega toate componentele din proiect

Cablajul va arata astfel:



Am conectat componentele conform lui:



Pinii conectati pentru fiecare componenta:

1. Conectare LCD 4×20 cu interfata IDC2:

- GND cu GND de pe Arduino (cu ajutorul Breadboard-ului)
- VCC cu 5V de pe Arduino (cu ajutorul Breadboard-ului)
- SDA cu A4 de pe Arduino
- SCL cu A5 de pe Arduino

2. Buzzer:

- GND cu GND de pe Arduino
- VCC cu D3

3. Cititor de card SD

- GND cu GND de pe Arduino
- VCC cu 5V de pe Arduino (cu ajutorul Breadboard-ului)
- MISO cu D12 de pe Ardino
- MOSI cu D11 de pe Arduino

- SCK cu D13 de pe Arduino
- CS cu D10 de pe Arduino

# Software Design

Pentru a dezvolta software-ul am folosit Arduino IDE.

Bibliotecile folosite sunt:

- LiquidCrystal_I2C pentru afişare pe ecranul LCD
- SPI şi SD pentru a putea scrie data pe cardul SD
- pitches.h pentru note muzicale predefinite plăcute
- bitmaps.h pentru a defini dinozaurul, cactusul şi pasărea ca nişte caractere, şi a le randa pe un singur pătrăţel de pe ecran

În funcţia de setup, am iniţializat ecranul, iar apoi pinul pentru buton ca INPUT, cel pentru buzzer şi cel pentru modulul de card SD ca OUTPUT.

Jocul are un splash screen ce este randat înainte să se apese butonul. In funcţia de loop, se verifică constant dacă butonl se apasă sau nu. Când acesta este apăsat, variabila bool splashScreen ia valoare fals, iar jocul începe. Jucătorul controlează prin buton când sare dinozaurul. Acesta este mereu plasat la poziţia 0 pe hartă. Pentru a face delimitarea între cadre, am folosit funcţia delay astfel: delay(200 - (score * 2)). În acest mod, cu cât scorul este mai mare, cu atât jocul se mişcă mai repede.

### Salt

Dinozaurul sare când jucătorul apasă butonul. Prin sărit ne referim la faptul că el va înainta pe rândul 2 în loc de rândul 3. Am făcut saltul sa îl imite cât mai aproape pe cel dintr-ul joc video real. Astfel, saltul are o durata predefinită ce este obţinută prin folosirea funcţiei millis. Ea returnează numărul de milisecunde care a trecut de când a fost pornit programul. Astfel, când butonul apăsat reţinem valoarea funcţiei millis şi mutăm jucătorul pe rândul 2. În loop verificăm dacă valoarea actuală a funcţiei millis minus valoarea reţinută în momentul efectuării saltului este mai mare decât durata saltului 500.

Totuşi jucătorul poate sări constant ceea ce nu este bine. Pentru a rezolva problema, am implementat un cooldown pentru salt. Astfel, în loop se va face încă o verificare: dacă millis minus valoarea ei reţinută în momentul când s-a efectuat saltul este mai mare decât valoarea cooldown-ului 1000. Valoarea aceasta trebuie sa fie mai mare decât cea care denotă lungimea saltului, deoarece timpul de cooldown efectiv în care jucătorul este pe pomânt şi nu poate să sară este 1000 - 500 = 500. Folosim o variabilă bool canJump care este falsă de când se efectuează saltul până când condiţia descrisă devine adevărată.

### Obstacole

La fiecare cadru obstacolele se apropie de dinozaur, prin randarea lor la poziţia anterioară minus 1. Obstacolele sunt de 2 feluri: cactuşii, care sunt pe nivelul pământului şi păsările, care sunt pe nivelul saltului. Pentru a face jocul sa fie echilibrat, am decis ca, la un moment dat, pe ecran pot sa fie maxim 2 cactuşi şi o pasăre. Astfel, trei variabile care reţin poziţia fiecăreia dintre ele. Am încercat să simulez generarea random a obstacolelor. Astfel, când există doar un cactus în scenă, cu cât acesta

se apropie de dinozaur, cu atât şansa să vina în scena cel de al doilea cactus creşte. Dacă un cactus ajunge pe poziţia 1, lângă jucător (care este mereu pe 0), fără ca un al doilea cactus sa fi intrat în scenă, acesta va apărea garantat pentru a menţine ciclul. Pentru pasăre, am decis să existe o şansă de 15% ca ea să apară în orice cadru în care nu există deja şi în care cei doi cactuşi au poziţia mai mică decât 15 şi poziţia lor nu este cea mai din dreapta (cazul în care tocmai au apărut). Această condiţie asigură că pasărea nu poate apărea deasupra unui cactus, astfel făcând rezolvarea imposibilă.

Pentru a testa coloziunea dintre dinozaur şi obstacole, în fiecare cadru se verifică:

- pentru cactuşi → dacă dinozaurul este pe pământ şi pozitia cactusului este 0
- pentru pasăre → dacă dinozaurul este în salt şi poziţia păsării este 0

Scorul este incrementat de fiecare dată când un obstacol este depăşit cu succes.

**Muzică**

Am definit global un vector care reţine notele melodiei în ordine şi unul care reţine durata fiecărei note. Cântecul este redat doar în timpul jocului, nu şi pe splash screen sau game over screen. Pentru a face o pauză între fiecare notă, m-am folosit de delay-ul ce separă cadrele. Astfel, este redată nota curentă, se incrementează variabila ce reţine nota curentă sau resetează dacă am ajuns la finalul melodiei, se alepează delay şi se apelează noTone. Pentru ca delay-ul devine din ce în ce mai mic, melodia va fi cântată mai repede pe măsură ce avansează jocul.

**Salvare scoruri**

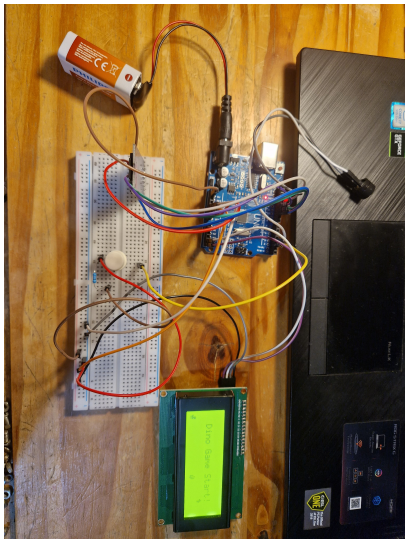
Atunci când o coliziune este detectată, variabila bool gameOver devine true şi apare ecranul de game over, împreună cu scorul obţinut. Se deschide fişierul scores.txt configurat pentru scriere de pe cardul SD. Se scrie sub formatul “Score: x”, apoi se închide fişierul. La apăsarea butonului, se revine la splash screen, şi astfel jocul se repetă.

**Resurse:**

- https://gist.github.com/mikeputnam/2820675
- https://www.engineersgarage.com/how-to-create-custom-characters-on-lcd-using-arduino-part-5-49/

# Rezultate Obţinute

Proiectul cu bateria conectată:



Proiectul aşezat într-o cutie astfel încât doar ecranul, buzzer-ul şi butonul să fie vizibile:


## Concluzii

Nu am mai folosit Arduino deloc înainte de a realiza acest proiect, deci pot spune că am învăţat multe de la comanda pieselor potrivite până la scrierea codului.

Majoritatea lucrurilor nu am mers din prima şi a fost nevoie de mult căutat pe Internet şi debugging. Unul dintre primele lucruri la care m-am blocat a fost faptul ca trebuia să înşurubez puţin şurubul de pe spatele I2C-ului de pe ecran, pentru a vedea bine.

Totuşi să duc până la capăt proiectul a fost satisfăcător, mai ales când pot să îl folosesc la final.

Link demo: https://youtu.be/PXyI-jHPegI

## Download

Arhiva cu codul: savin_ana_bianca_dino_endless_runner.zip

# Bibliografie/Resurse

Resurse hardware:


- https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/
- https://docs.arduino.cc/built-in-examples/digital/Button/
- https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/


Resurse software:


- https://gist.github.com/mikeputnam/2820675
- https://www.engineersgarage.com/how-to-create-custom-characters-on-lcd-using-arduino-part-5-49/


Export to PDF