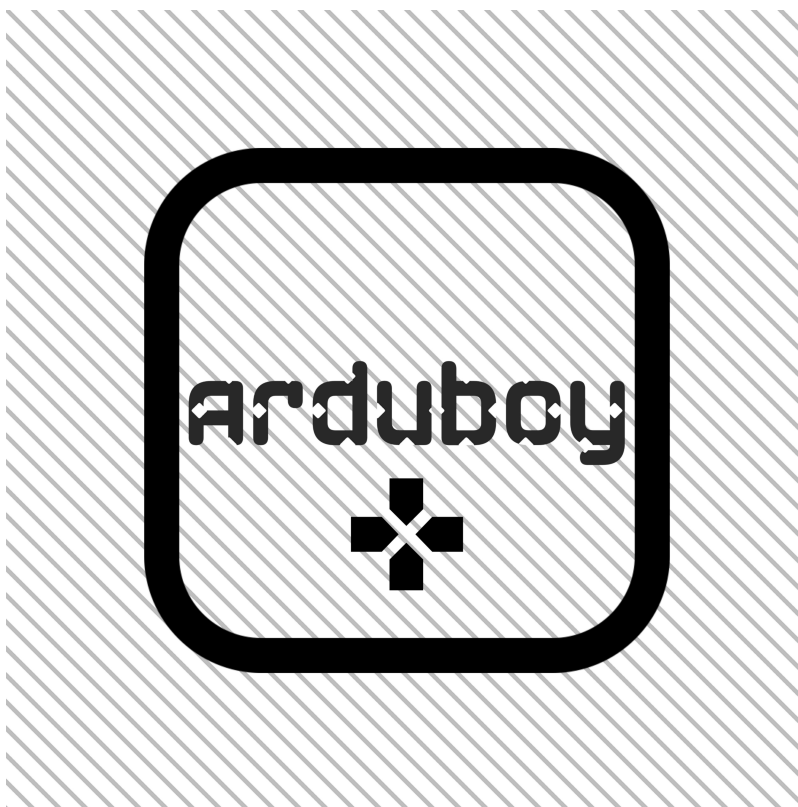


Mini consola portabila

Introducere



Proiectul meu este o implementare destul de simpla a unei mini console portabile, utilizand conceptele studiate in laboratoarele de PM. Sistemul este similar cu o consola retro (seamana cu un Gameboy ca performanta) si dispune de un afisaj color. Sistemul vine cu 3 jocuri preinstalate (Pong, 2 Player Pong si TRex Jump), cu posibilitatea de a mai scrie si alte jocuri (in limita memoriei placii Arduino UNO).

Descriere generală



Jucatorului i se va prezenta un ecran principal in care se va boota mereu consola. El va putea folosi butoanele de navigatie pentru a decide in ce joc sa intre, dupa care jocul va incepe sa ruleze. Jucatorul nu este aruncat direct in joc, consola va afisa un ecran de intro pentru fiecare joc in parte.

Soft-ul nu va incarca initial decat bootloader-ul de jocuri, player-ul urmand sa selecteze ulterior ce joc

sa fie initiat. Odata ce un joc a fost selectat trebuie sa dam un reboot consolei pentru a putea selecta alt joc.

Pong

Jucatorul va putea controla o platforma pentru a incerca sa tina mingea in ring cat de mult posibil. El se va juca cu computer-ul si va incerca sa trimita mingea in spatele platformei lui. Pentru a deplasa platforma in sus sau in jos, jucatorul va folosi butoanele A si B.

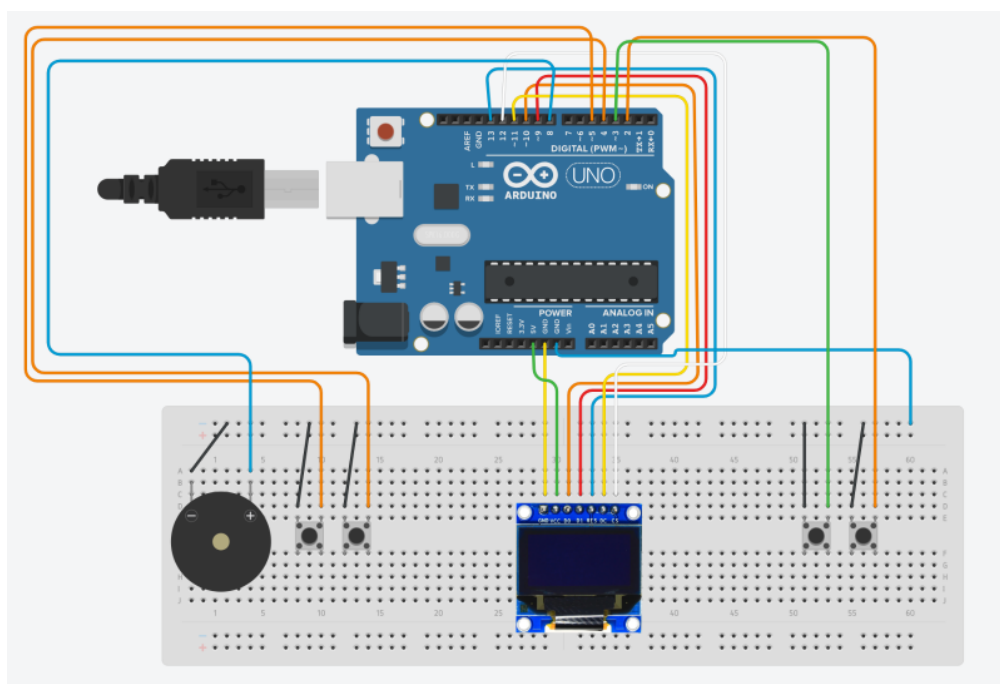
2 Player Pong

Logica este destul de similara cu cea de la Pong-ul obisnuit, doar ca acum platforma din stanga nu va mai fi controlata de computer, ci de un alt jucator, folosind butoanele de navigatie din stanga pentru a deplasa platforma in sus sau in jos.

TRex Jump

TRex Jump este un joc in care controlam un dinozaur ce merge la nesfarsit si trebuie in traseul lui sa se fereasca de cactusi. Pentru aceasta jucatorul trebuie sa apese A pentru a face TRex-ul sa sara, moment in care jucatorul primeste un punct la scor.

Hardware Design



Pentru proiectarea consolei, am folosit urmatoarele componente:

- Placa Arduino UNO R3
- Ecran OLED 128 x 64 prin SPI
- Patru push buttons
- Buzzer 3V

Software Design

Pentru implementarea proiectului, pentru dispunerea graficii jocurilor pe ecranul meu am folosit bibliotecile "**Adafruit_GFX.h**" si "**Adafruit_SSD1306.h**" impreuna cu cea de **SPI** pentru ca ecranul sa poata sa comunice cu placa Arduino.

Initializare si Ecranul Principal



Inainte de afisarea vreunui meniu pe ecran, in functia de **setup()** imi initializez o variabila pentru display pentru afisarea informatiilor pe ecran. Mai initializez pinii folositi pentru butoane si buzzer si incep sa afisez ecranul principal unde imi este prezentata o lista de jocuri din care pot selecta ce sa joc. Pentru afisarea jocurilor imi aloc dinamic o lista in care retin denumirile fiecarui joc implementat in proiect. Sistemul va ramane blocat in acest ecran cat timp jucatorul se va decide ce sa joace. Jucatorul va putea apasa oricand butoanele de navigatie stanga-dreapta pentru a cicla prin lista de jocuri. Intern lista functioneaza similar cu un hashmap, am o variabila pe care incrementez/decrementez cand apas stanga/dreapta iar pe ecran afisez denumirea jocului aflata in lista la index-ul determinat de variabila. Pentru a controla butonul si pentru a ma asigura ca nu se inregistreaza decat o singura apasare cand ciclez prin lista folosesc functia **millis()** in combinatie cu o serie de if-uri in care ma asigur ca in 100 de milisecunde nu se poate inregistra decat o singura apasare efectiva de buton. Cand va apasa B jocul afisat curent va incepe sa ruleze. Pentru a nu pune overhead inutil pe procesorul placutei ecranul principal nu va rula decat o singura data intr-un loop in functia de **setup()** ci nu in cea de **loop()** efectiv. Din acest motiv daca dorim sa schimbam jocul pe care l-am ales trebuie sa repornim cu totul consola. Pentru redarea sunetelor la orice moment de timp in jocuri am folosit functie **tone()**.

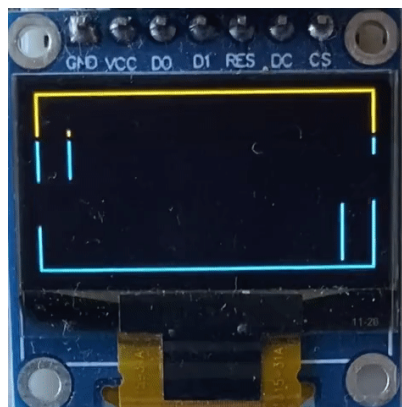
In functie de valoarea ramasa in variabila de decizie stim cu siguranta ce joc a fost selectat. In functia de **loop()** pe baza acestei valori se face tranzitia de la ecranul principal catre jocul selectat.

Jocul de Pong



Initial sistemul va afisa un ecran generic de intro al jocului, urmand ca pentru inceperea jocului sa fie nevoie sa fie apasat butonul A. Dupa ce se apasa A ecranul va incepe sa deseneze automat pe "conturul" lui linii ce vor reprezenta terenul de joc. Vor fi desenate doua platforme si o minge la niste coordonate prestabilite dupa care mingea va incepe sa se miste pe ecran. Intregul algoritm din spatele jocului sta in implementarea coliziunilor pentru a reda efectul jocului de Pong. Au fost implementate coliziuni intre platforme si pereti, intre minge si pereti si intre minge si platforme. Practic, la fiecare moment de timp se va calcula o noua pozitie pentru minge si se va redesena pe ecran (daca coordonatele sunt valide). Cand se intampina o coliziune pe vreuna dintre axe cu pereti sau platforme, directia de deplasare pe acea axa pentru minge se inverseaza. Dupa ce calculez coordonatele ma folosesc de functia **drawPixel()** din biblioteca Adafruit cu noile coordonate ca parametru, pentru a afisa mingea la noua pozitie.

Pentru platforme procesul este similar, pentru platforma jucatorului folosesc actiunea butoanelor A si B pentru a incrementa/decrementa coordonatele platformei. Pentru computer, procesul este simplist doar va cicla la infinit intre a urca si cobora platforma in functie de ce perete loveste ultima oara. Pentru afisarea graficilor cand se actualizeaza coordonatele in acest caz folosesc functia de biblioteca **drawFastVLine()**. In orice moment de timp cand mingea se loveste de vreo platforma se aude un sunet scurt de coliziune.



La final pentru ca toate aceste actualizari sa se poata produca apelez functia **display()**.

Jocul 2 Player Pong



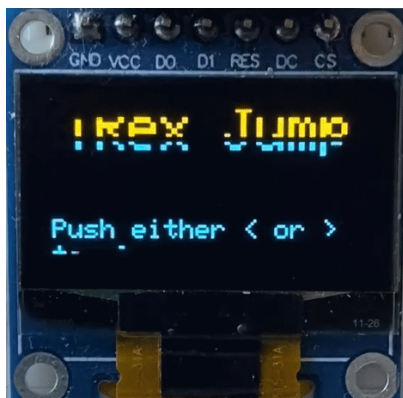
Ca logica jocul este aproape identic, cu singura diferenta fiind in platforma din stanga ecranului, care acum nu mai cicleaza la infinit intre sus si jos, ci este controlata de alt jucator de butoanele din stanga de navigatie pe aceeasi logica ca la platforma din dreapta, in rest implementarea nu are diferente.

Jocul TRex Jump



La fel ca la Pong, jocul va afisa initial un ecran generic de intro unde jucatorul este invitat sa apese oricare din butoanele de navigatie pentru a face TRex-ul sa mearga. :)

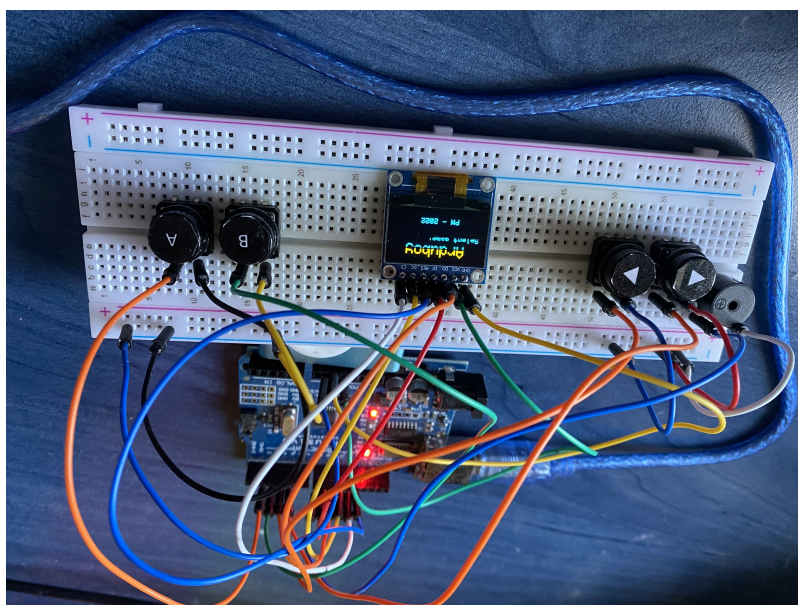
Inainte de a incepe jocul efectiv initializez coordonatele si dimensiunile pentru dinozaur si pentru cactusi. Pentru a genera grafica pentru obiecte folosesc niste char arrays cu un design bitmap ce va fi folosit in functia de biblioteca **drawBitmap()** impreuna cu coordonatele si dimensiunea obiectului respectiv. Pe tot parcursul jocului cactusii se vor genera random, 2 la un anumit moment de timp si se vor deplasa spre TRex. In momentul in care coordonatele cactusilor arata ca ar trebui sa paraseasca ecranul acestia sunt regenerati in partea opusa a ecranului (la un anumit moment de timp) doar ca fiecare design in parte pentru cactus va fi regenerat random, pentru a da impresia de realism.

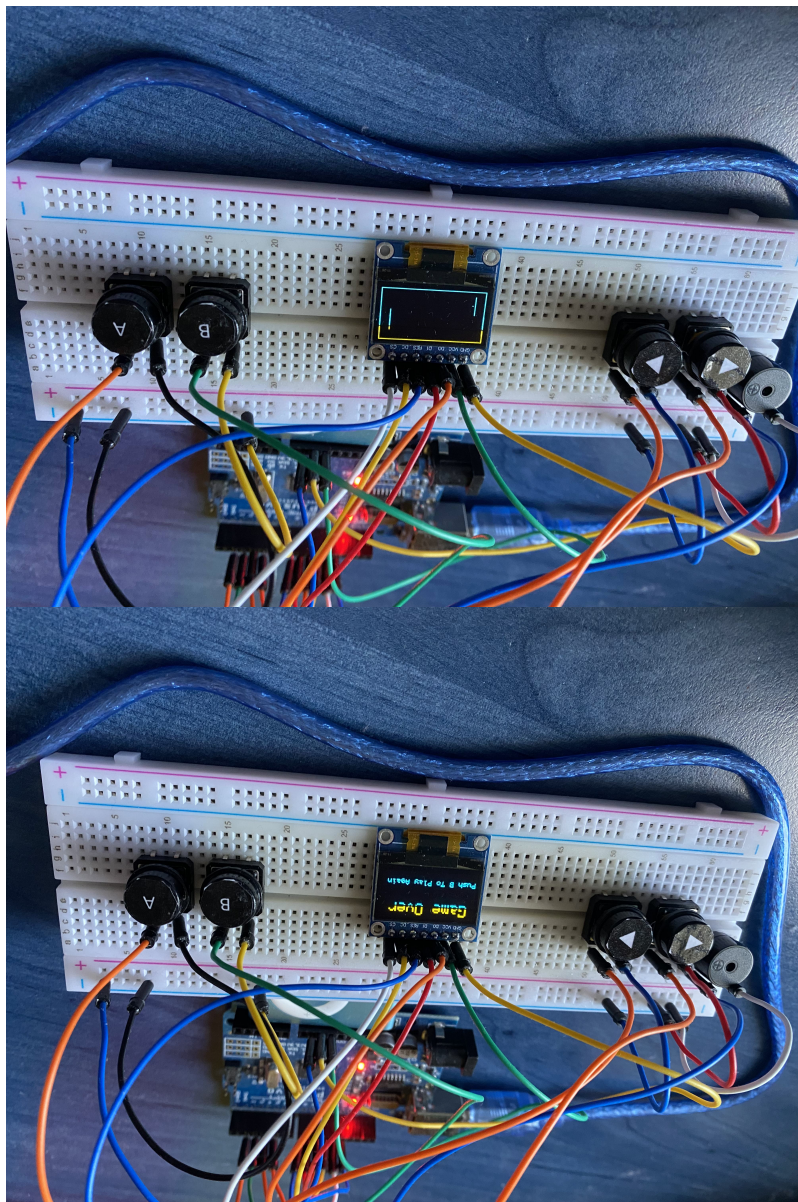


In orice moment de timp daca jucatorul va apasa pe butonul A TRex-ul va sari iar daca reuseste sa scape din calea cactusului (nu este trigger-uita coliziunea dintre ei) atunci scorul se incrementeaza cu 1. Atunci cand dinozaurul sare peste un cactus, buzzer-ul canta un sunet scurt. Cand se intampla sa se loveasca de vreun cactus jocul se opreste si se afiseaza un ecran cu mesajul "Game Over" impreuna cu scorul obtinut si de posibilitatea de a reincepe jocul daca se apasa butonul B. Cand apare acest ecran se aude si un sunet mai lung si adanc specific unui joc ce se afla in starea de game over.



Rezultate Obținute





Un demo se poate gasi aici:

[Arduboy Demo](#)

Concluzii

Prin acest proiect am inteles cat de important este sa optimizezi cat mai mult codul (in general). Niciodata nu stim ce resurse vor avea alte persoane asa ca este bine sa optimizam tot ce se poate atunci cand scriem un program...si da...este important sa ne asiguram ca legam firele cum trebuie.

Download

[Cod Sursa - Mini Consola Portabila](#)

Jurnal

Bibliografie/Resurse

[Arduino - How to press button only once](#)

[How to use Adafruit displays](#)

[TRex bitmap designs](#)

[How to output sounds in the buzzer](#)

[How to use PROGMEM for TRex game](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/amocanu/catalin.barus>



Last update: **2022/05/23 18:22**