

Rhythm Game

- Nume: Vârban Ana-Maria
- Grupa: 336CB

Introducere

- Proiectul consta in implementarea unui joc de ritm in stil similar jocurilor precum TapTap, Piano Tiles sau SuperStar. In mod specific, jucatorul trebuie sa apese butoanele disponibile pe ritmul notelor muzicale ce vor aparea pe un ecran. Astfel, jucatorul va obtine un scor ce va depinde de numarul de note apasate la timp si acuratetea cu care au fost apasate acestea. Jocul se va sfarsi la finalul melodiei iar scorul va aparea pe ecran.

Descriere generală

- Pe display-ul LCD va aparea un prompt de start pentru a putea incepe jocul. Jucatorul va porni jocul folosind butoanele de control.
- Notele vor aparea pe ecran alaturi de scorul obtinut de jucator pana atunci, iar acesta va trebui sa apese butoanele de joc pe ritmul melodiei pentru a obtine un scor cat mai bun.
- Jocul se incheie la terminarea melodiei iar scorul obtinut este afisat pe ecran.
- Jucatorul poate alege dintre optiunile de exit si replay. Optiunea exit va duce catre ecranul de start iar optiunea replay va restarta melodia. Selectia se va face folosind butoanele de control.

Schema Bloc



Hardware Design

Piese folosite:

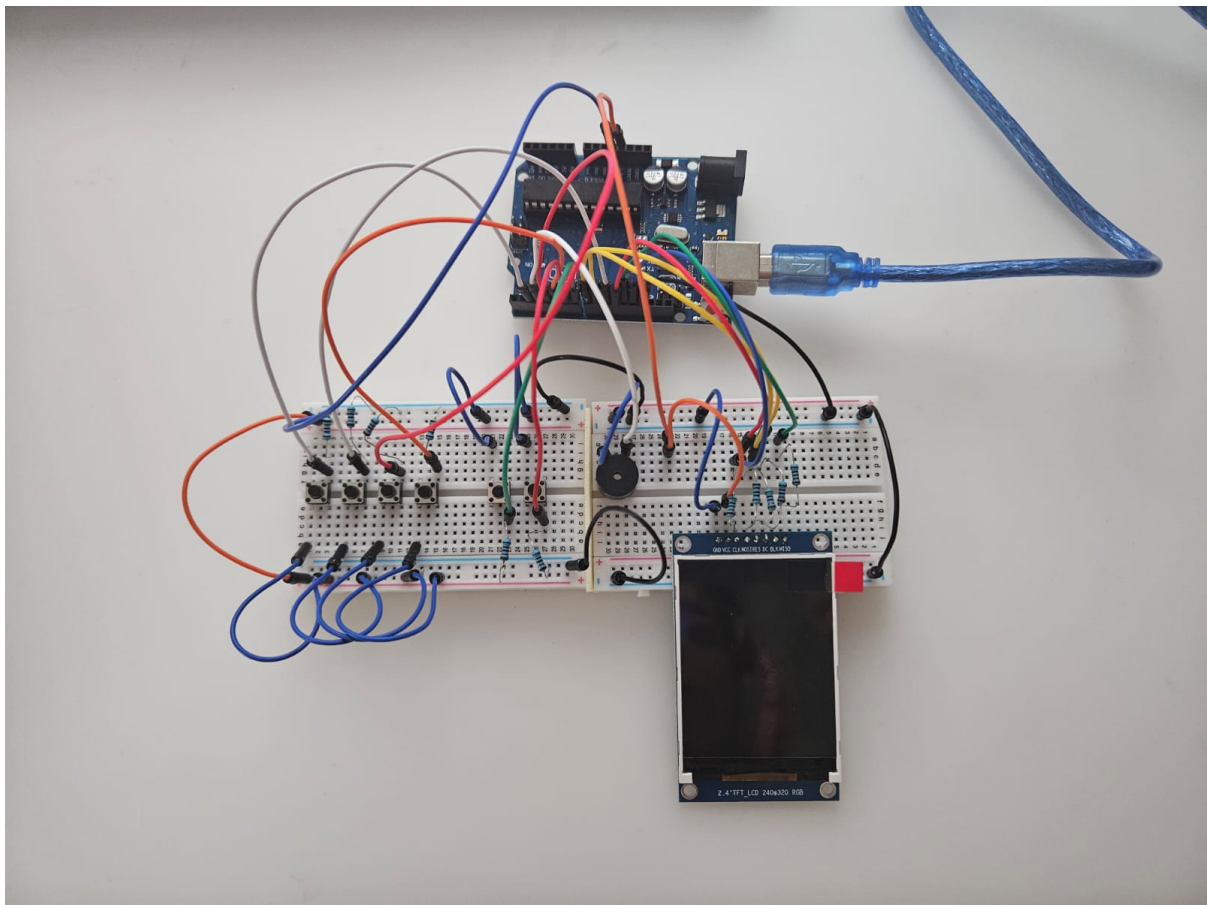
- * Arduino Uno (ATMega328p)
- * Ecran LCD (LCD SPI 2.4" ILI9341)
- * Butoane (4 butoane pentru joc, 2 pentru control)
- * Buzzer pasiv
- * Rezistori 10k
- * Fire de legatura

* Breadboard

Schema Electrica



Implementarea Hardware



- După cum se poate observa, am așezat cele **4 butoane ce vor fi folosite pentru joc** în stânga breadboard-ului, iar cele **2 butoane de control** în partea dreaptă. Butoanele sunt conectate alături de câte un rezistor de 10K ce acționează ca rezistența de pull down. Cele 6 butoane sunt conectate la pinii digitali de pe placuta **Arduino Uno**. **Ecranul LCD** este așezat pe breadboard alături de rezistori de 10K, conform indicațiilor distribuitorului. Aceste rezistențe previn arderea display-ului. Pinii folosiți pentru LCD sunt pinii digitali 8, 9, 11-13. **Buzzerul** este conectat la unul dintre pinii digitali PWM ai placutei Arduino, acesta nefiind inserat cu un rezistor pentru un sunet mai puternic.

Software Design

Deoarece notele ce vor apărea pe ecran sunt singurele elemente ce se vor mișca, acestea sunt

definite de clasa **NoteCircle**. Aceasta pastreaza atribute precum pozitia curenta a notei, pozitia precedenta a notei, daca nota este sau nu pe ecran, daca nota este sau nu un *blank*, spatiul in care se afla sau dimensiunea si metode precum *getteri* si *setteri*, respectiv o metoda de update a pozitiei. Pentru afisarea acestora pe ecran, am declarat global un array (**circles**) de 30 de elemente, dintre care 15 vor fi *blank*. Am ales sa folosesc note *blank* (note ce nu vor fi afisate) pentru a ma asigura ca notele nu sunt ingramadite pe ecran.

Melodia ce va fi redata la apasarea notelor este un fragment din Korobeiniki (aka The Tetris song) si este reprezentata printr-un array declarat global.

In cadrul functiei **setup()**, pornesc interfata seriala pentru a putea comunica cu display-ul, setez pinii digitali aferenti butoanelor si buzzer-ului si asignez toate notele de la indecsi impari in array-ul **circles** ca fiind note *blank*.

In cadrul functiei **loop()**, se determina starea jocului. Jocul se poate afla la un moment de timp intr-una dintre 3 stari predefinite: MENU, PLAY sau SCORE.

In starea MENU, se afiseaza pe ecran titlul jocului (ARDUINO RHYTHM GAME) si un mesaj indemnand jucatorul sa apese butonul (asociat firului) verde pentru a incepe jocul. Afisarea acestor elemente este facuta cu ajutorul functiei **displayMenu()**.

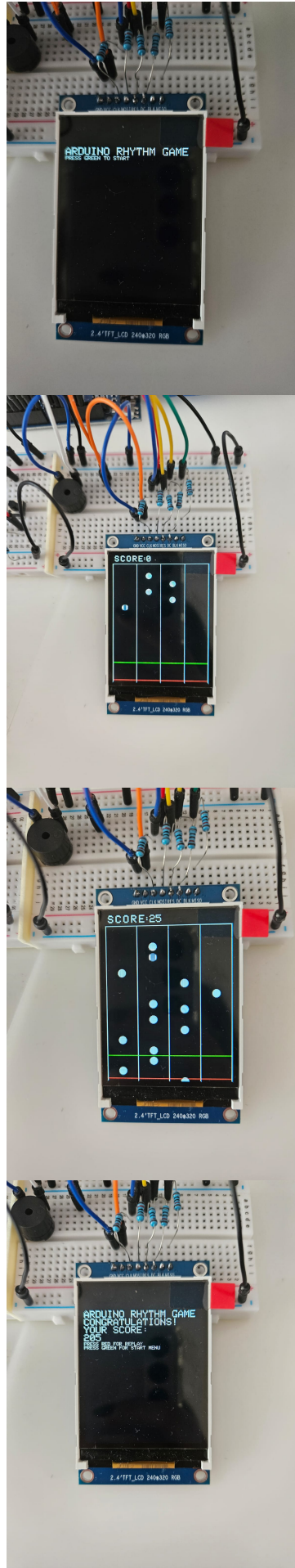
Odata apasat butonul de start, jocul se va afla in starea PLAY iar pe ecran vor aparea cele 4 spatii in care vor aparea notele. Notele vor ajunge la baza ecranului, in zona delimitata de o linie verde si una rosie, unde, daca jucatorul apasa butonul corespunzator, scorul va creste iar nota aferenta din cantec va fi redata sonor. Pentru verificarea localizarii corecte a notei se va verifica doar coordonata *y*, deoarece odata afisata, o nota nu isi va modifica coordonata *x* cat timp este pe ecran. Pentru verificarea apasarii butoanelor, se citesc valori de pe **pinii digitali** aferenti acestora. Pentru redarea notei se va mentine indexul curent din array-ul de note, acesta fiind pasat functiei **playSong(int note)**, care va reda sunetul corespunzator. Jocul se termina cand toate notele din cantec au fost redade. Pentru afisarea spatiului de joc ma folosesc de functia **drawPlayField()**, iar pentru logica din timpul jocului, de functia **play()**

Odata terminat cantecul, jocul va trece in starea SCORE, unde se va afisa pe ecran scorul obtinut. De aici, jucatorul poate apasa fie butonul verde, pentru a reveni la meniul de start, fie butonul rosu, pentru a juca din nou. De afisarea acestora se ocupa functia **displayFinalScore()**.

Scrierea si incarcarea codului pe placuta Arduino au fost facute folosind *Arduino IDE*. De asemenea, am folosit bibliotecile *SPI.h*, pentru comunicarea cu display-ul si *Adafruit_GFX.h*, respectiv *Adafruit_ILI9341.h* pentru afisarea elementelor grafice pe display. Codul aferent redarii sunetelor este preluat din proiectul *Arduino Songs*, de care ne-am folosit si la laboratorul 3.

Rezultate Obținute

Prin cele de mai sus am obtinut un joculet minimalist de ritm. Din pacate, limitarile componentelor au dus la un gameplay relativ lent. Mai jos se gaseste link-ul catre un demo.



Demo Rhythm Game

Concluzii

Pe masura ce am implementat diversele functionalitati, m-am lovit de limitarile componentelor, precum viteza de transfer a datelor de la placuta la ecranul LCD, afisarea pe ecran a notelor si reimprospatarea scorului, ce se face foarte des, motiv pentru care acesta apare sa clipeasca. Acestea au dus la un joc cu o complexitate foarte redusa si la un gameplay relativ lent.

In ciuda aspectelor mai neplacute, a fost foarte imbucurator sa vad rezultatele pe ecran. Creerea de la 0 a ceva tangibil si functional a fost o experienta unica, de care nu am mai avut parte pana acum. Desi proiectul este unul rudimentar, sunt foarte mandra de el.

Download

[Download Rhythm Game](#)

Jurnal

- 25 aprilie 2023 : Am ales tema si am creat pagina wiki pentru proiect
- 1 mai 2023: Am completat pagina wiki cu o descriere a proiectului si un outline al design-ului hardware
- 2 mai 2023: Am comandat piesele necesare
- 9 mai 2023: Am lipit pinii display-ului la laborator
- 13 mai 2023: Am asamblat componentele si am finalizat design-ul hardware, am testat functionalitatile de baza ale componentelor
- 24 mai 2023: Am implementat meniul de start, afisarea spatiului de joc si afisarea scorului
- 26 mai 2023: Am implementat caderea notelor, incrementarea scorului si redarea notei la apasarea corecta a butoanelor
- 30 mai 2023: PM Fair!

Bibliografie/Resurse

- [Cod exemplu display LCD](#)
- [Biblioteca display LCD](#)
- [Cod Arduino Songs](#)
- [PM Lab 2](#)
- [PM Lab 3](#)
- [PM Lab 5](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/iotelea/rhythm-game>



Last update: **2023/05/29 14:15**