

Memory Game

Introducere

Memory Game este o reinterpretare digitală a clasicului joc de memorie Simon Says.

Scopul acestui proiect este de a oferi o experiență interactivă și distractivă, care stimulează memoria și atenția jucătorului.

Ideea de la care am pornit: Inspirația provine de la conceptul simplu și captivant al jocului Simon Says și am dorit să îl aduc în era digitală, folosind o plăcuță Arduino și componente hardware accesibile.

Utilitate: Este util pentru alții deoarece oferă o modalitate distractivă și educativă de petrecere a timpului liber, stimulând memoria și atenția jucătorilor de toate vârstele. Acest proiect reprezintă o oportunitate de a îmi dezvolta abilitățile în ingineria hardware și software, experimentând cu tehnologii noi și obținând satisfacție din rezultatele muncii depuse.

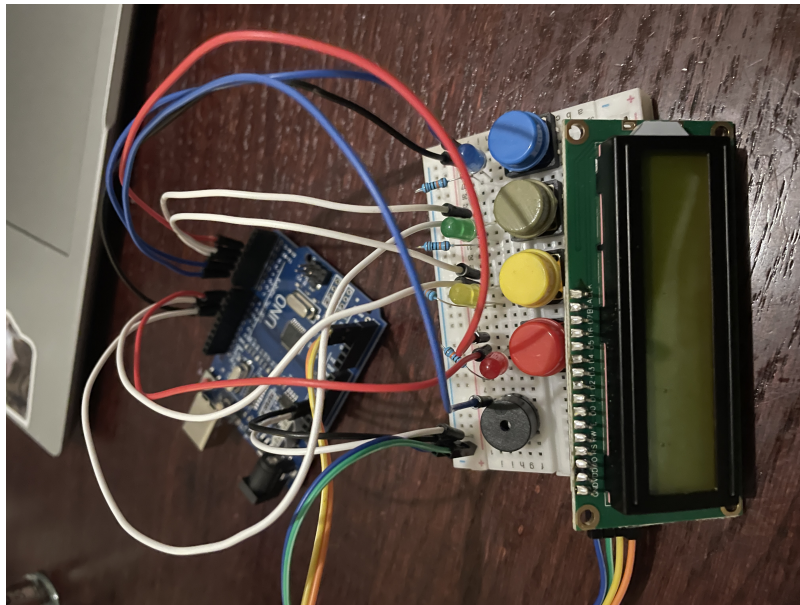
Descriere generală

Proiectul constă într-un joc de memorie bazat pe conceptul popular Simon Says. Jucătorul interacționează cu dispozitivul folosind butoanele pentru a reproduce secvențele luminoase generate aleatoriu. Arduino Uno servește ca unitate centrală de procesare, primind intrările de la butoane și generând ieșiri corespunzătoare pentru buzzer, display-ul LCD I2C și LED-uri. Scopul jocului este de a ajunge cât mai departe prin memorarea și reproducerea corectă a secvențelor în creștere de semnale.

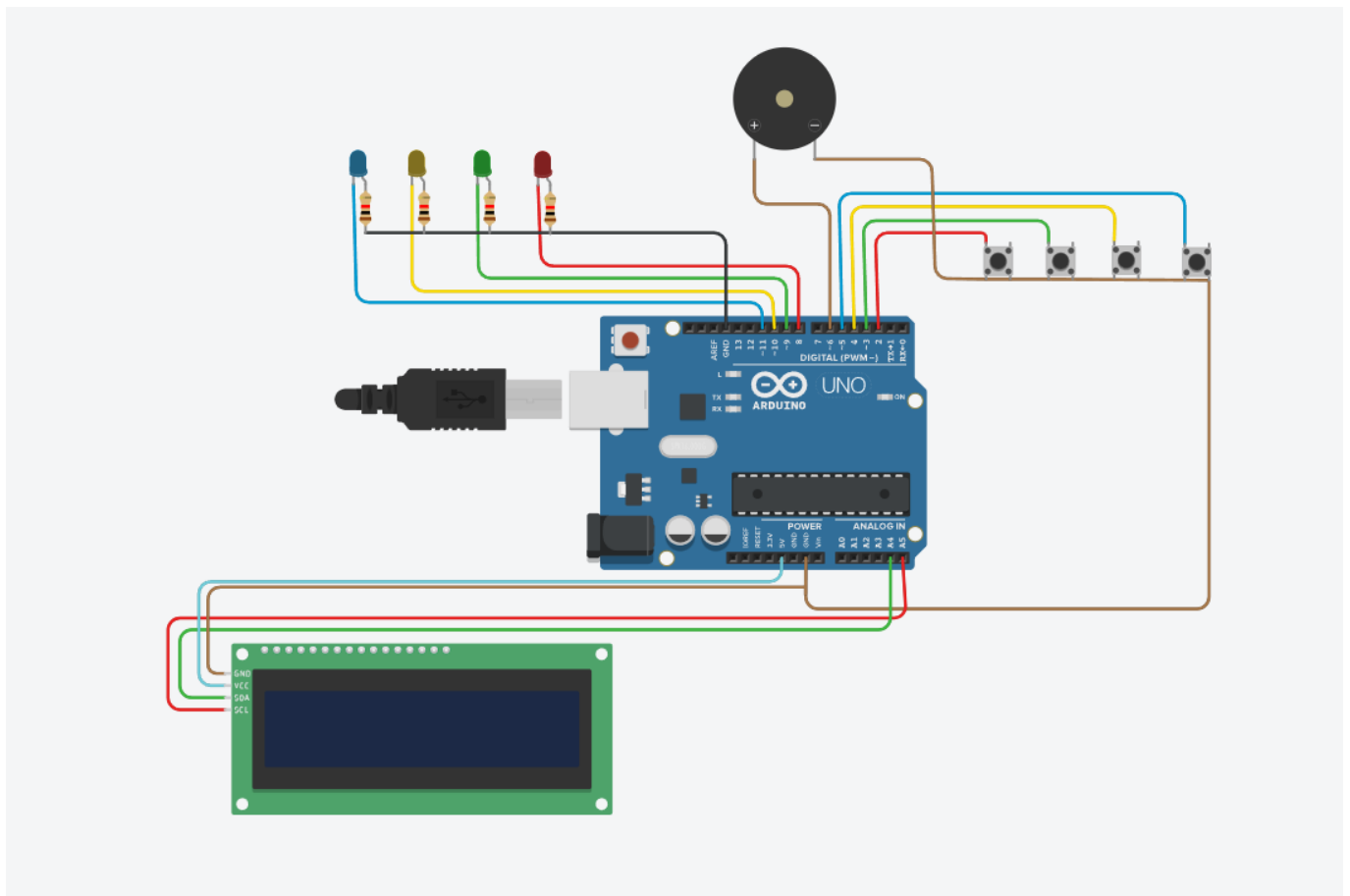


Componentele utilizate în proiect sunt următoarele:

- Arduino Uno
- Display LCD 16x2
- Modul LCD I2C
- Buton de apăsare
- Rezistor 100R
- LED-uri
- Buzzer
- Fire de conexiune
- Breadboard
- Fire jumper tată-tată
- Fire jumper tată-mamă



Schema electrică:



Software Design

1. Descrierea codului aplicației (firmware)

Mediu de dezvoltare

Pentru dezvoltarea acestui proiect am utilizat PlatformIO, un mediu de dezvoltare integrat (IDE) bazat pe Visual Studio Code. PlatformIO oferă suport pentru o gamă largă de plăci de dezvoltare și microcontrolere, inclusiv Arduino Uno, pe care l-am folosit în acest proiect. De asemenea, oferă instrumente avansate pentru depanare, gestionarea bibliotecilor și integrare continuă, ceea ce facilitează dezvoltarea și testarea firmware-ului.

2. Librării și surse 3rd-party

În cadrul acestui proiect am folosit următoarele librării externe:

LiquidCrystal_I2C - Pentru a facilita interacțiunea cu afișajul LCD folosind protocolul I2C.

Wire - O bibliotecă standard Arduino pentru comunicarea I2C, necesară pentru funcționarea bibliotecii LiquidCrystal_I2C.

Arduino.h - Biblioteca de bază a platformei Arduino, care oferă funcții esențiale pentru manipularea pinii, temporizări și comunicare serială.

3. Algoritmi și structuri implementate

Algoritmul jocului de memorare:

- Generarea secvențelor de LED-uri într-un mod aleator.
- Afișarea secvențelor pe LED-uri și LCD.
- Verificarea secvenței introduse de utilizator pentru a determina dacă este corectă sau greșită.

Vectori (array):

- `buttons[]` și `leds[]` pentru a stoca pinii butoanelor și LED-urilor.
- `bt_simonSaid[]` și `led_simonSaid[]` pentru a stoca secvențele generate și secvențele introduse de utilizator.

Variabile booleene:

- `lost` pentru a determina dacă utilizatorul a pierdut jocul.

4. Funcții principale:

`setup()`:

- Inițializează comunicația serială.
- Configurează pinii butoanelor ca intrări cu rezistență de pull-up.
- Configurează pinii LED-urilor și buzzerului ca ieșiri.
- Inițializează LCD-ul și afișează mesajul de bun venit.
- Setează o valoare inițială pentru generatorul de numere aleatorii.

`loop()`:

- Gestionează stadiile jocului folosind un switch statement:
 - Stadiul 0: Așteaptă ca utilizatorul să apese butonul roșu pentru a începe jocul.
 - Stadiul 1: Afișează nivelul curent și generează o secvență nouă.
 - Stadiul 2: Așteaptă ca utilizatorul să introducă secvența.
 - Stadiul 3: Verifică secvența introdusă de utilizator.
 - Stadiul 4: Afișează mesajul de pierdere și resetează jocul.

- Stadiul 5: Afișează mesajul de câștig și trece la următorul nivel.

playBuzzer(int x):

- Redă un sunet pe buzzer pentru a indica apăsarea unui buton.

Rezultate Obținute

Funcționalitatea jocului: Implementarea inițială a jocului de memorare a întâmpinat dificultăți în ceea ce privește generarea și verificarea secvențelor de LED-uri. Afișajul LCD nu a fost corect, multe informații nefiind vizibile din cauza firelor de calitate slabă.

Depanarea și testarea: Testarea și depanarea codului au evidențiat probleme de inițializare și configurare a componentelor hardware. Integrarea componentelor hardware (butoane, LED-uri, buzzer, LCD) nu a fost realizată complet cu succes, ceea ce a dus la nefuncționarea corectă a jocului.

Concluzii

Proiectul a întâmpinat diverse probleme care au afectat funcționarea sa corectă. Dificultățile principale au fost legate de inițializarea și configurarea corectă a componentelor hardware și de afișarea adecvată a informațiilor pe LCD. Deși proiectul nu a funcționat conform așteptărilor inițiale, procesul de dezvoltare a fost educativ. Am învățat lecții valoroase despre necesitatea unei structuri clare a codului, importanța testării riguroase și a depanării continue. Această experiențe va fi de mare ajutor în realizarea viitoarelor proiecte.

Download

[baluta_teodora.zip](#)

Jurnal

25 aprilie: Selectarea temei proiectului

2 mai: Crearea paginii de wiki și elaborarea schemei bloc

9 mai: Plasarea comenzilor pentru piesele necesare

16 mai: Implementarea componentelor hardware

22 mai: Dezvoltarea și implementarea software-ului

Bibliografie/Resurse

<https://www.youtube.com/watch?v=BJ93XCcD858>

https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

https://www.youtube.com/watch?v=gj-H_agfd6U&t=3s

<https://projecthub.arduino.cc/Jerepondumie/make-an-arduino-memory-game-c9c093>

<https://www.instructables.com/Arduino-Simple-Memory-Game/>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/alucaci/teodora.baluta2107>



Last update: **2024/05/23 11:01**