

Tic-tac-toe Game

Autor

Roșculete Anca-Maria

Introducere

Proiectul constă într-un joc simplu de X și 0, realizat pe Arduino UNO, folosind ca display un LCD SPI de 1.8", iar interacțiunea cu jocul se va permite printr-un joystick.

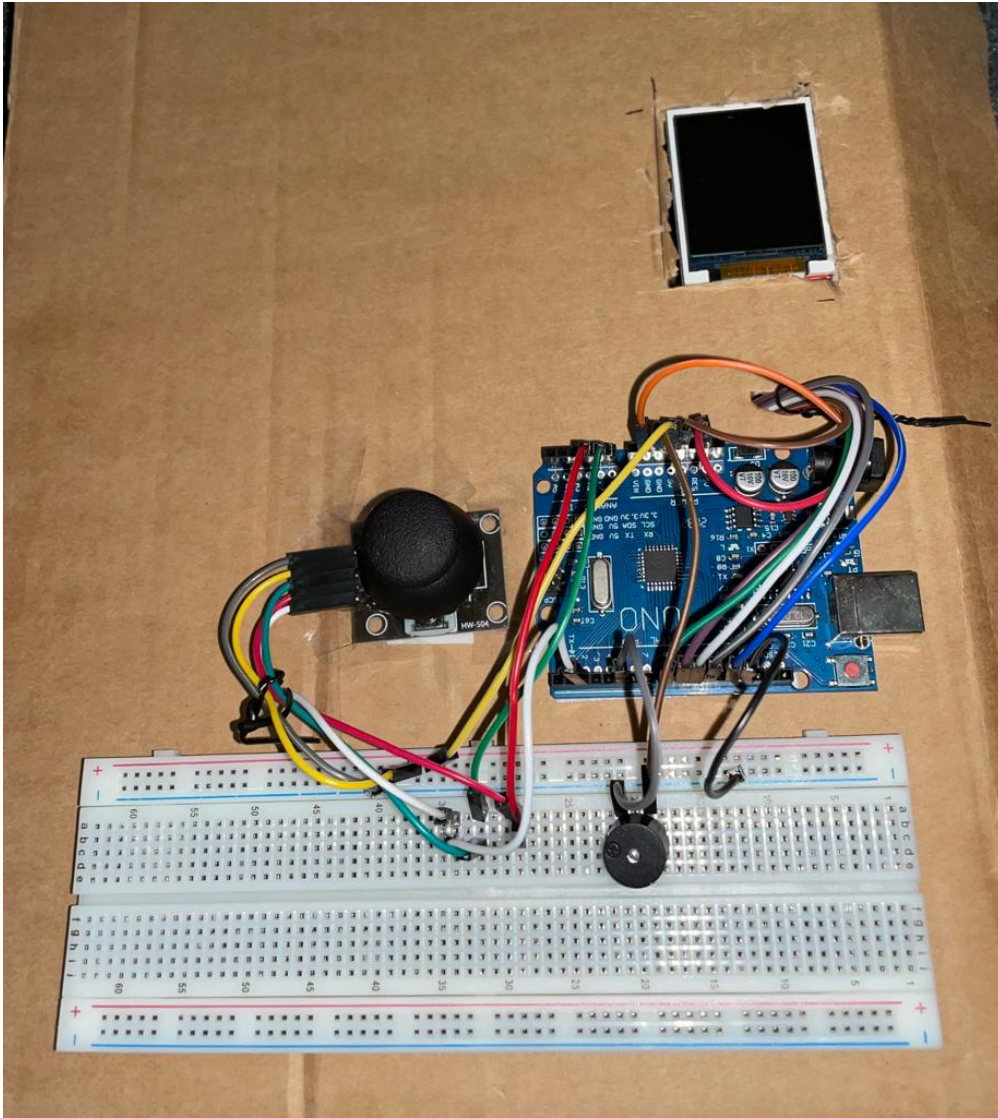
Rolul acestui proiect este de familiarizare cu Arduino prin implementarea unui joc simplu și cunoscut tuturor.

Descriere generală

Jucătorul va vedea pe ecran un careu și semnul X cu care se va putea deplasa între pătrățelele careului cu ajutorul mișcărilor realizate prin joystick. Odată ce se decide unde vrea să își plaseze semnul (X), el va confirma alegerea prin apăsarea joystick-ului. Dacă alegerea este pe o poziție goală, buzzerul va emite un sunet simplu, dar dacă încearcă să pună pe o poziție deja ocupată, buzzerul va scoate un sunet mai intens, iar jucătorul va trebui să facă altă mutare. Acesta va avea ca adversar "computerul"(O), care folosește un algoritm Minimax pentru a decide următoarea mutare a sa.

Schema bloc





Hardware Design

Listă de piese:

- Arduino Uno
- Joystick
- Modul LCD SPI (128x160)
- Breadboard
- Buzzer pasiv 5V
- Fire de legătură

Schema electrică



Software Design

Descrierea codului aplicației:

- mediu de dezvoltare: Arduino IDE
- librării și surse 3rd-party (pentru LCD):

1. Adafruit GFX
 2. Adafruit ST7735
 3. SPI.h
- algoritmi și structuri de date:
 1. logică simplă (if, while, for, etc.)
 2. structuri de bază C: variabile, constante, etc.
 3. algoritm Minimax pentru mutările calculatorului(O)
 - surse și funcții implementate:
 1. funcția **setup()** inițializează pozițiile pe axe ale liniilor careului, pozițiile pentru X și O în fiecare careu, diferite variabile care marchează statusul jocului și poziția curentă a jucătorului, pinii pentru joystick, buzzer, de asemenea desenează careul inițial și poziția inițială la care se afla jucătorul X. Mai mult, buzzerul va produce un zgomot care marchează începutul jocului. Este inițializată și o matrice 3×3 pentru a ști ce poziții sunt ocupate și de către cine.
 2. funcția **loop()** afișează de fiecare dată noile informații și citește pinii pentru axele X și Y de la joystick și mută X-ul conform datelor primite de la joystick. Aici, citind pinul SW de la joystick, se confirmă mutarea dacă este validă și se dă prin funcția tone() semnal la buzzer pentru a scoate sunet.
 3. funcțiile **drawSymbolX(currentX, currentY)** **drawSymbolO(currentX, currentY)** desenează simbolul X/O la poziția din matrice primită ca parametrii la funcție.
 4. funcția **drawBackground()** o folosesc deseori de-a lungul programului pentru a redesena careul și simbolurile X/O deja prezente
 5. funcția **isMovesLeft(matrix[3][3])** verifică dacă mai sunt locuri libere în matrice pentru a ști dacă s-a finalizat jocul.
 6. funcția **evaluate(matrix[3][3])** este folosită în algoritmul Minimax pentru a verifica victoria unuia dintre jucători.
 7. funcția **minimax(matrix[3][3], depth, isMax)** implementează efectiv algoritmul, folosindu-se și de funcția findBestMove.
 8. funcția **checkGameState()** o folosesc în funcția loop() pentru a verifica dacă a câștigat cineva, dacă e remiză sau încă nu s-a terminat jocul.

Rezultate obținute

[Cod sursă](#)

[Video Demo](#)

Concluzii

Proiectul pentru PM a reprezentat o provocare, dar chiar mi-a plăcut să descopăr ce pot realiza. Partea de hardware nu mi s-a părut complicată, în cadrul acest proiect software-ul era cel ce necesita mai mult timp. Sunt sigură că pot îmbunătăți algoritmul pentru mutările calculatorului și se mai pot adăuga noi feature-uri. Proiectul meu m-a ajutat să mă familiarizez foarte tare cu lucrul cu LCD-ul și constituie un punct de plecare pentru alte aplicații pe care le-aș mai putea realiza pe acesta (lucrul cu LCD-ul și desenarea obiectelor a fost partea care mi-a luat cel mai mult, dar odată ce m-am obișnuit lucrurile au mers repede).

Bibliografie/Resurse

- <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf>
- https://exploreembedded.com/wiki/Analog_Joystick_with_Arduino
- <https://www.youtube.com/watch?v=xDgwxYMmhGI>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/tac-tac-toe_ar



Last update: **2021/05/29 09:18**