

X si O

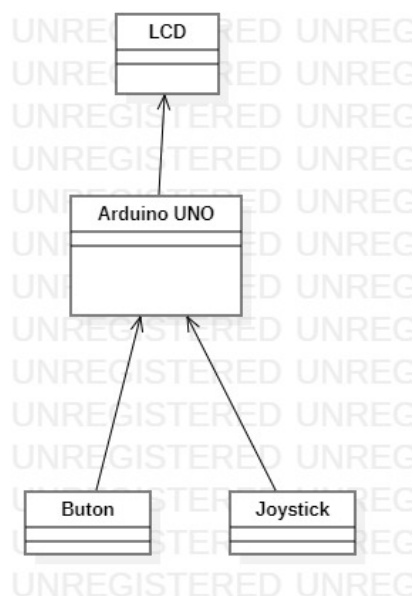
Nume : Stefan Andreea Bianca

Grupa : 334CB

Introducere

X si O este un joc clasic apreciat de oameni de toate varstele. Este un joc pentru doi jucatori, jucat pe o matrice de 3×3. Scopul jocului este sa fii primul jucător care obține de trei ori simbolul sau într-o linie, fie orizontal, vertical sau diagonal. În acest proiect, vom implementa X si O pe o placă Arduino folosind un LCD, joystick si buton. Jocul va fi afișat pe LCD, iar jucătorii vor folosi butonul si joystickul pentru a face mutările.

Descriere generală



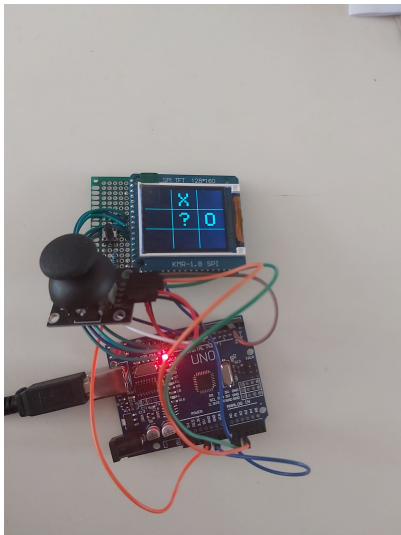
Jucatorii vor utiliza pe rand joystickul pentru a naviga tabla de joc 3×3 si vor folosi butonul pentru a confirma mutarea (dispozitivul tine cont de numarul runde si introduce automat simbolul corect). Tabla de joc este verificata la sfarsitul fiecărei runde daca a fost jucata o mutare castigatoare (jucatorul a completat un rand de 3 cu ultima sa mutare) si de asemenea se verifica daca mai exista spatii goale, in cazul in care nu s-a detectat o victorie, iar tabla este plina, partida va fi declarata remiza. La finalul partidei se va afisa un mesaj care corespunde modului de incheiere (jucatorul 1 / 2 a

castigat sau a avut loc o remiza).

Hardware Design

Lista de piese

- Arduino Uno
- Breadboard
- LCD
- Fire de legatura
- Joystick
- Buton
- Rezistente



Software Design

Libraries:

```
SPI.h  
TFT.h
```

The game logic is implemented in the code using a 2D array to represent the game grid. Each cell in the grid can have one of the following values:

```
"-1": Empty cell  
"0": Player 0's mark  
"1": Player X's mark
```

"2": Temporary mark (used during player X's turn)

The game checks for a winning condition by examining each row, column, and diagonal. If a winning condition is found, the game displays the winner and resets the grid for a new round. If all cells are filled and no winner is found, the game declares a draw and resets.

Functions:

`void next():`

Updates the previous position and value variables to keep track of the previous move in the game.

`bool isWin():`

Checks if there is a winning condition in the game. It examines each row, column, and diagonal to determine if all three cells have the same non- empty value (either 0 or 1). If a winning condition is found, it returns true, otherwise, false.

`void resetGame():`

Resets the game by reinitializing all the necessary variables and the game grid. It sets the grid cells to -1 (empty) and resets the position and counter variables.

`void Enter():`

Handles the player's move when the button is pressed. It assigns the appropriate mark (X or O) to the selected cell on the grid based on the round counter. It then checks for a winning condition or a draw and displays the corresponding message. If the game is not over, it resets the necessary variables and increments the round counter.

`void Round():`

Handles each round of the game. It updates the grid based on the joystick input (left, right, up, or down) by changing the current position and updating the temporary mark (2) accordingly. It redraws the grid after each move. When the button is pressed, it calls the Enter() function to process the player's move and increments the round counter.

`void drawX(int x, int y):`

Draws the X mark on the display at the specified coordinates (x, y).

`void drawO(int x, int y):`

Draws the O mark on the display at the specified coordinates (x, y).

`void drawtemp(int x, int y):`

Draws a temporary mark ? on the display at the specified coordinates (x, y). This mark is used during the players turn before they confirm their move.

`void draw(int c, int i, int j):`

Draws the appropriate mark (X, O, or ?) on the display at the specified grid position (i, j) based on the value stored in the game grid.

void drawAll():

Redraws the entire grid on the display by iterating through the game grid and calling the draw() function for each cell.

void displayWinner():

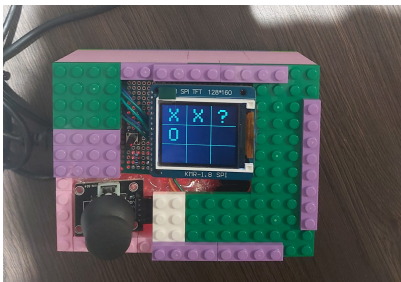
Displays the winner (X or O) on the display when a winning condition is met. It shows the corresponding message for a certain period of time before resetting the game.

void displayDraw():

Displays a draw message on the display when the game ends in a draw. It shows the message for a certain period of time before resetting the game.

Rezultate Obținute

demo: <https://youtube.com/shorts/6qapzdg4-10?feature=share>



Concluzii

It's a simple implementation of a Tic-Tac-Toe game using an Arduino board and a TFT display. Players can navigate the grid using a joystick and place their marks (X or O) by pressing a button. The game checks for a winning condition or a draw after each move. It provides an interactive and enjoyable gaming experience with the Arduino platform.

Download

Arhiva:

[stefan_bianca.zip](#)

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite:

https://exploreembedded.com/wiki/Analog_JoyStick_with_Arduino

<https://docs.arduino.cc/retired/getting-started-guides/TFT>

<https://create.arduino.cc/projecthub/electropeak/ultimate-beginner-s-guide-to-run-tft-lcd-displays-by-arduino-081006>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2023/drtranca/x_si_o



Last update: **2023/05/29 22:47**