

Introduction:

The game of Tic Tac Toe has been known since Ancient Egypt and is undoubtedly one of the most popular of all time. However, the methods for playing it vary greatly, thus in this project I set out to create a device that can be used for Tic Tac Toe.

General Description:

The grid is represented by a matrix of RGB LEDs, with each player having a color (rather than a symbol - "X" or "O") that he selects from a set of predefined colors (red, green, blue, yellow, magenta, and cyan) before the start of the game. Using the buttons ("←", "→", and "ok"), players can select the box in which they enter their color. At each stage of the game, the score and several indicators are presented on an LCD monitor. Throughout the game, the grid's edge pulses in the color of the player whose turn it is.



Software desing:

The code starts by including the libraries required for controlling NeoPixel LEDs (Adafruit_NeoPixel) and LCD screens (LiquidCrystal).

There are constants and global variables specified. Pin assignments, LED count, grid size, button cooldown time, and variables that monitor the current player, player colors, grid status, and player scores are examples of these.

The code then declares function prototypes for all of the functions utilized.

To initialize the hardware and set up the game, the setup() method is invoked. It sets up the NeoPixel strip and LCD display, as well as calling the restart() and pickColor() methods.

The main game loop (loop()) gets started. It keeps running the game logic till the game is over.

To check for button presses and manage the relevant actions, the handleRightButton(), handleLeftButton(), and handleOkButton() routines are invoked. These functions handle the input from the player in order to position their marker in the grid.

The colorBorder() function is used to change the color of the LED strip border based on the player currently in use.

To update the LED strip to match the current state of the grid, the colorGrid() function is invoked.

The code determines whether the game is over by comparing the sum of player scores to the highest possible score. If the game has concluded, it runs the rejoice() function if there is a winner or the draw() function if there is a tie, and then calls the restart() function to restart the game.

The gameStart() function is used to display a welcome message and alert the first player that the game is about to begin.

The restart() function returns the game to its initial condition. It resets player scores, clears the grid, and resets the current player to 1.

The pickColor() function shows a message on the LCD to inform the players of the colors they have been allotted (red and green).

The updateCurrentPlayerFactor() function uses the current time to update the animation factor of the current player's color. It produces a pulsating effect on the edge of the LED strip.

The blinkCurrentPlayer() function highlights the spots on the LED strip where the current player has placed their marker for a brief period of time. To represent the player's movement, it generates a flashing effect.

By changing the value of the currentPlayer variable, the switchCurrentPlayer() function passes the turn to the next player.

To announce the winning player, the celebrate() method shows a message on the LCD.

To signify a tie game, the draw() method shows a message on the LCD.

Code Source: [tictactoe.rar](#)

Bibliografy/Resources

<https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use>

<https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/avaduva/tictactoe>



Last update: **2023/05/29 14:27**