

TicTacToe on the PIC32 Microprocessor

Objective and Requirements

My goal was to implement the classic game of tic tac toe using the pic32 microprocessor screen, along with several input buttons. I also wished for the game to support three different game modes, one for a single game played, and options for Bo3 and Bo5's. The game was to use the screen on the chipKit as output.

Solution

I decided to use the functions for manipulating the screen which come with the lab files. To display the game board. I used the `display_string` functions to create the actual playing board, and ascii character X's and O's to represent the playing pieces. Furthermore I used character pointers to represent the squares on which the game is to be played. This was a way of isolating the characters we wish to manipulate throughout the progress of the game from the ones that merely represent the game board architecture.

The way I chose to implement the game menu where one is to choose desired gamemode was also done using the `display_string` method. There is both a visual token and a corresponding integer which is incremented and decremented by the 3rd push button. When the 4th push button is pressed, the menu function returns with current value of the token integer, signifying the gamemode. This integer corresponds to the amount of games that need to be won by each player to win the tournament.

This number is then sent to a function which repeatedly creates a tic tac toe game, receives the eventual winner, and decreases the number corresponding to player1 or player2. When either of these is zero, signifying that one player has won, the result is printed to the screen.

The actual game is controlled by a function which repeatedly, until a winner is detected for this specific game, allows the two players, in turn, select empty squares. The position which is currently highlighted is controlled by the 4 switches. 2 for controlling x direction, and 2 for controlling y direction.

Verification

My verification model was rather informal actually. However, since the game is so simple, and I was able to continuously use the board display to see what was going on with my game, it was trivial to detect bugs and errors. I also tested each new game functionality as I implemented them. Thus, my verification method was built off of constant small tests, which was made possible by structuring the order in which I designed the game in a logical manner.

Contributions

Everything was done by yours truly.

Reflections

I can honestly say that I had fun building this game, and I am proud of what I have created. This is in spite of the fact that the game is incredibly boring and not very nice looking. This is because I feel like this was the first time I have been pushed to program in a way where I can make my own decisions, and where I feel like both my creativity and patience is being pushed. I feel like I have become much more confident in my programming skills, and I feel fairly confident with the C programming language. In short, I appreciate what this project accomplished for me, even though I am glad that it is over :).