CS120B Final Project: Simon

Collaborators: Jeremy O'neill, Gabe Cortes

Benjamin Li - bli020

Overview

Our team decided to do the game of Simon. Our goal was to imitate the old game of simon using our breadboard and software that we were presented in class. The purpose of the game is to test the user's memory. The board will play a sequence with a maximum length of 9, and the user must play the exact same sequence on the board.

Technologies

- Microcontroller
- Breadboard
- LCD display
- 4 LEDs
- 5 buttons
- Speaker

Design

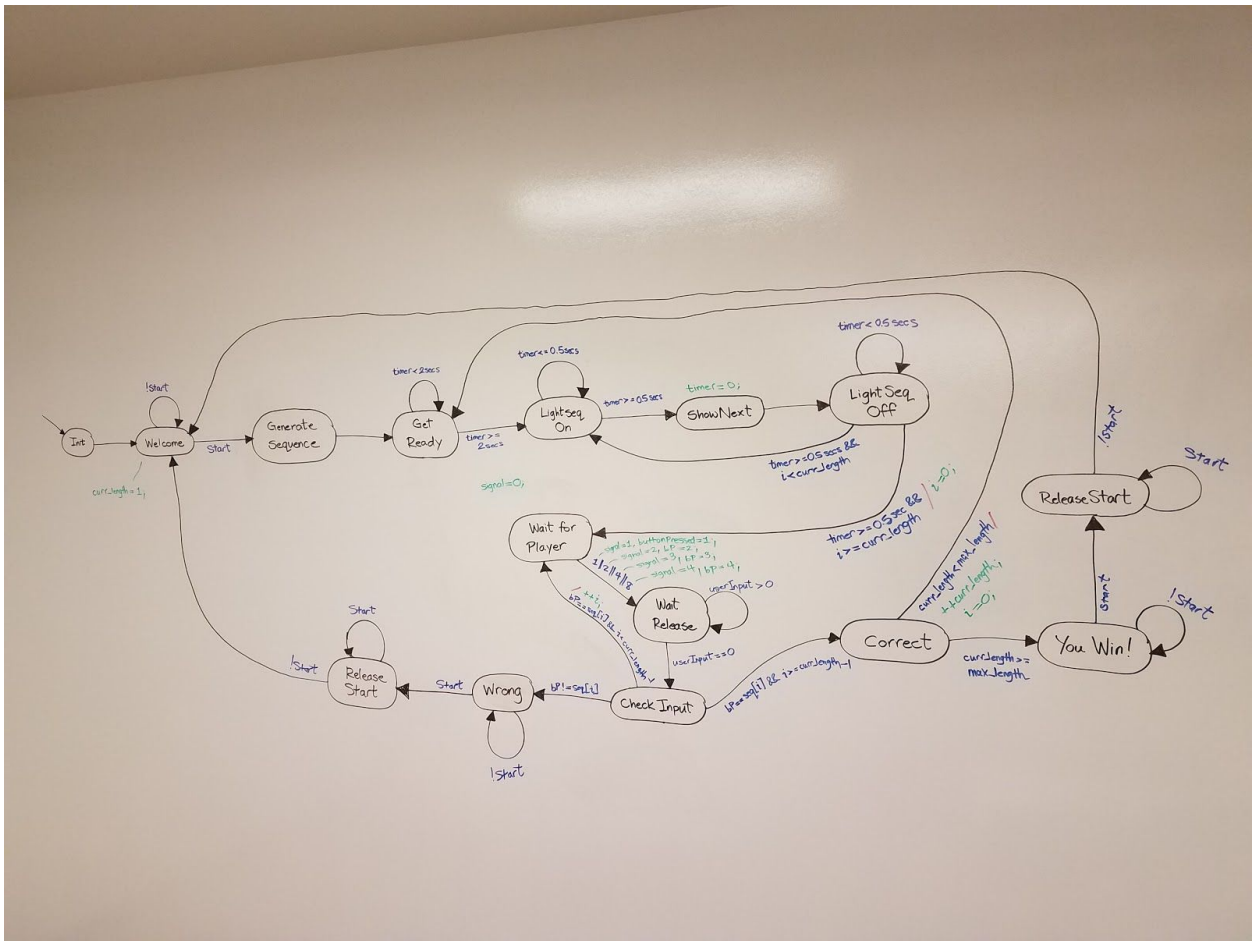
We started by making a concurrent state machine with 2 state machines. One of the state machines handled the methods of outputting such as lighting up the LEDS and making the sound after each light is lightened up. The second state machine takes care of the logic of the game such as taking in inputs and generating the sequence. It also will be there to determine whether the user won or lost the game.

First we designed the board to be able to take in a button press to start the game. After the button press we set a delay so the user could be able to get ready. Then the random sequence would begin.

We then had to be able to generate a specific sequence output it to the LEDS. The purpose of this is to let the user know which buttons to press. We first figured out how to output and make sure the sequence would end at 9. Then we implemented the rand() function to be able to output the sequence in a different order each time.

The next step we took was to have the machine be able to read in input from the buttons. After the input was read from the button, we compared it to the sequence to determine if it was right or wrong.

The final step was to determine what to do based on where the user was in the game. If it was right and not at the end of the sequence, the next sequence would be outputted. If it was wrong, then all the lights would light up simultaneously symbolizing error and allow them to restart. If it was right and at the end of the sequence, we would signal the user that they won and allow them to restart.



Above is the state machine that represents the logic of the process described above.

The other state machine that we made but were unable to take a picture of was the signal state machine. ITs sole purpose was to read which button was pressed or which signal was one and light up the corresponding light.

Responsibilities

We all worked as a group and divided and conquered the responsibilities of the code and the hardware. I was a part of the initial board design, testing the hardware and software, and exploit auditing(making sure we covered all our bases in the condition logic between transition states so that players cannot cheat or shortcut their way through the game).

Github: https://github.com/benbnli/simon
Video: https://www.youtube.com/watch?v=kLS5eUjj4XA