Final Project

Course: EE/CS 120B

Name : Jeremy Mah

Student ID: 860-990-799

Email: [jmah002@ucr.edu](mailto:jmah002@ucr.edu)

Partner: Miguel Vasquez

Email: [mvasq007@ucr.edu](mailto:mvasq007@ucr.edu)

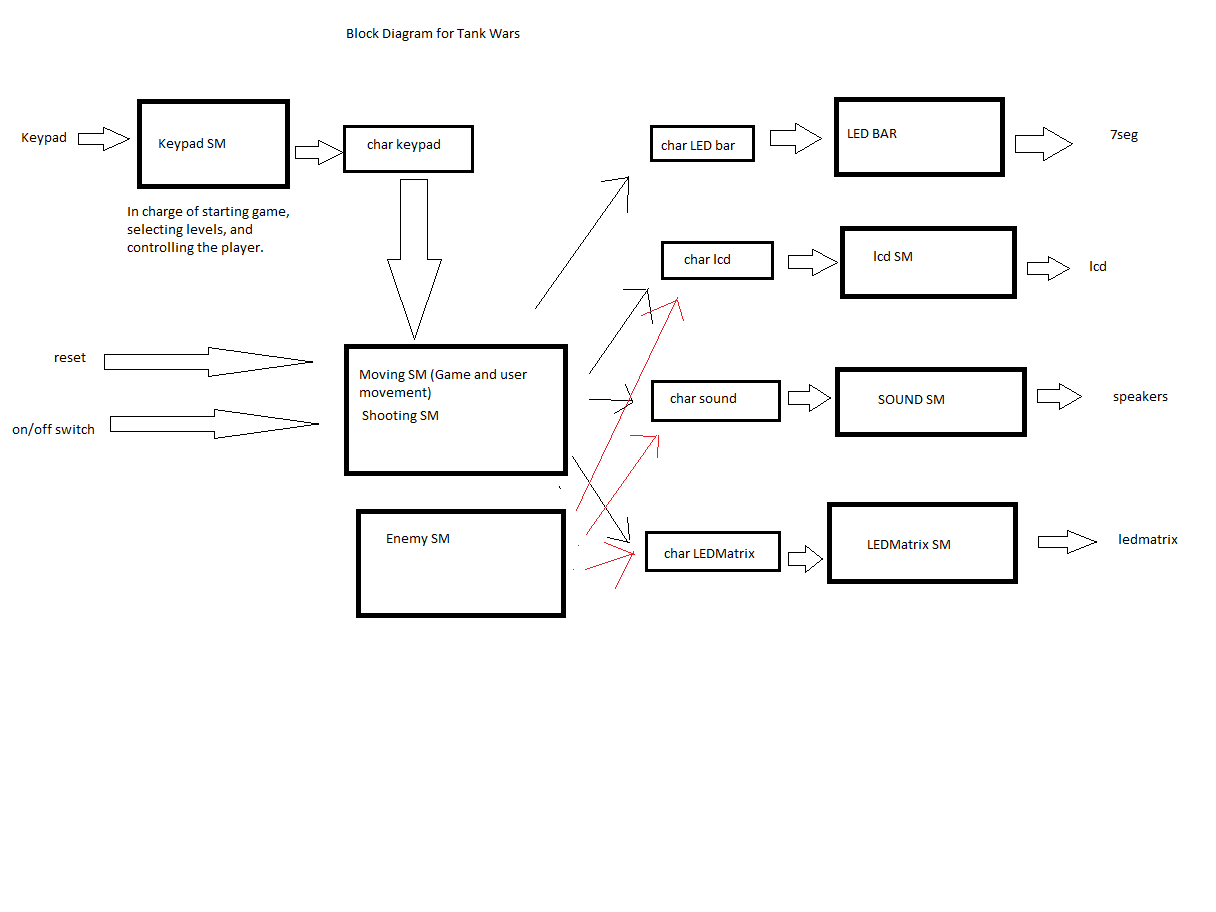
Date: 15 March 2013

Title : Tank Wars

We, Jeremy Mah and Miguel Vasquez, attest that this document

and the work described herein is entirely our own work

except where explicitly indicated otherwise.

High Level Description

This is my block diagram of the game. The switch and reset buttons starts the game and the only input is the keypad. The LED bar, LCD, Speaker, and LEDMatrix are all outputs from my shooting and moving state machines. The enemy state machine also uses LCD, Speaker, and LEDMatrix.

My game is Tank Wars and the goal of the game is to shoot down the enemy tank. The game starts when you turn on the game with the switch. The LCD Screen then asks you to press 1 or 2 to select difficulty, after pressing that, the LCD Screen will ask you to press 3 to start the gameplay. Then the LCD Screen displays “Game In Progress.” After the game ends, the LCD Screen will display “Reset to play”. The game starts with you, the blue dot spawning in the bottom right corner and the red dot spawning at the top left corner. You control the tank with the keypad. A is right, B is down, C is left, 6 is up, D is to shoot, and # is to shoot your special missile. You shoot in the direction you last moved. In both difficulties the enemy is programmed to go towards your location to kill you. Difficulty 1 isn’t necessarily easier, but it doesn’t directly charge at you. It has more random movement and shoots a lot more than difficulty 2. Difficulty 2 has significantly less moving and most moves are going to be aimed toward moving towards you or shooting towards you. The enemy in difficulty 2 also moves faster than the one in difficulty 1. If the enemy ever gets stuck after running into a green wall, he will shoot to destroy the wall. You are given two special missiles that the enemy doesn’t have when the game begins. These missiles are golden projectiles that can destroy multiple walls and will also destroy the enemy tank when contact is made. The downside to this special attack is that it moves slower than regular missiles. When a missile hits the wall, the enemy’s missile, and the enemy, the speaker buzzes to tell you something blew up.

User Guide :

The game:

To power the game on you have to press down the first switch. The LCD will then guide you through selecting the difficulty and ask when you would like to start the game. When the game starts, you will be controlling the blue dot and your objective is to shoot down the red dot. You spawn at the bottom right corner of the 8x16 RGB LED Matrix, and the enemy spawns at the top left. You are allowed to shoot and destroy walls to get to the enemy. You have two golden special missiles that are can have the same function as a regular missile except that it can shoot through multiple walls and can destroy the enemy tank if the enemy is hit by it. While your projectiles are being shot, you are allowed to move, so you can get in position to attack the enemy or so you can run away. If you shoot down the enemy and win the game, the bottom LED matrix will display a blue happy face, indicating that you won the game. If you lost the game, the LED matrix will display a red sad face. The LCD screen will then tell you to press reset if you want to play again.

User Controls:

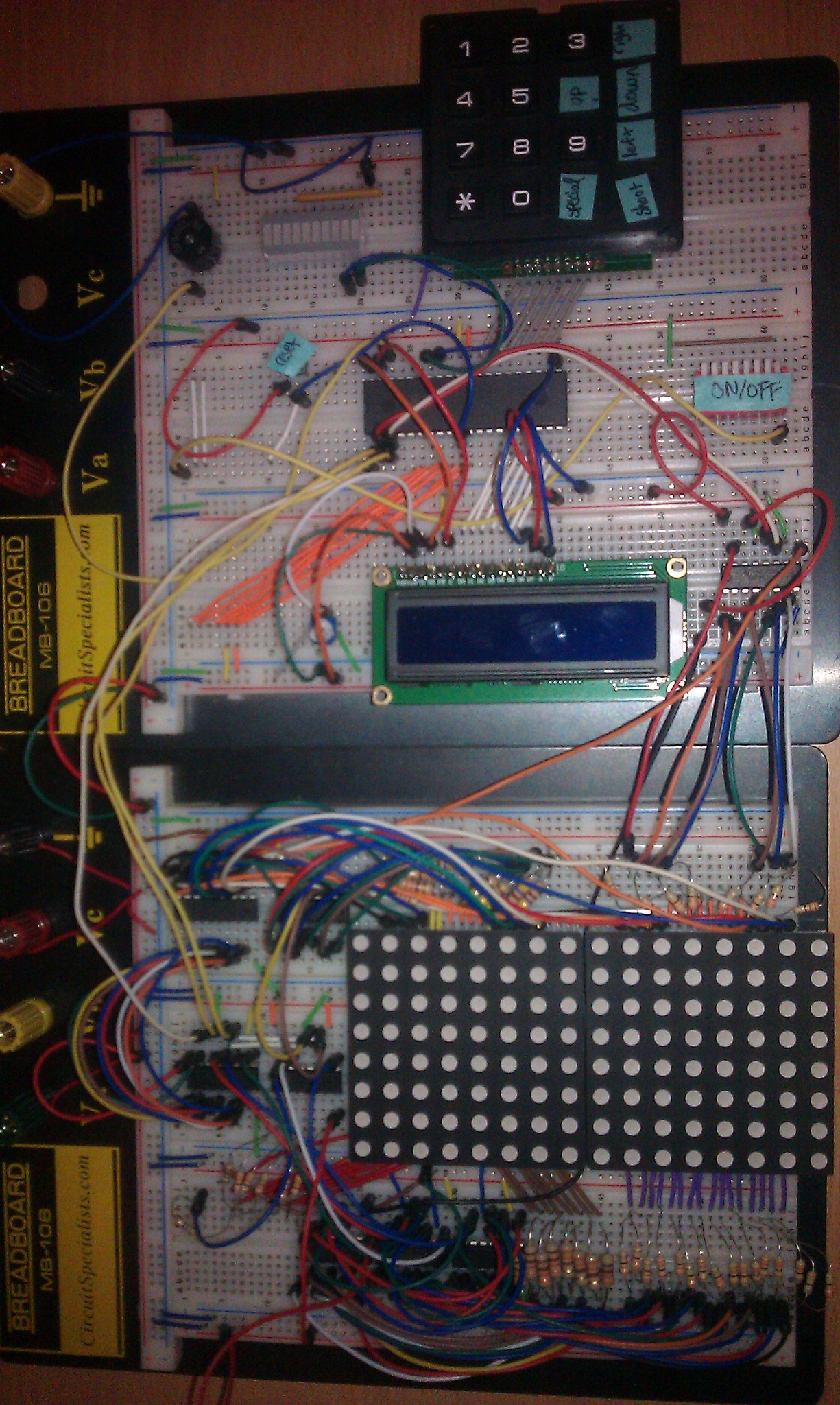
You will be using the Keypad to play the game. The buttons A is to move right, B is to move down, C is to move left, 6 is to move up, D is to shoot, and # is to shoot your special missile. The reset button resets the game and there is a switch where you can turn on/off the game.

Output:

I used 48 330 ohm resistors for all the lights in this matrix to prevent burning out. I also used a transistor array to prevent dimming problems. The speaker buzzes when something is hit and the LED is wired to a resistor bar.

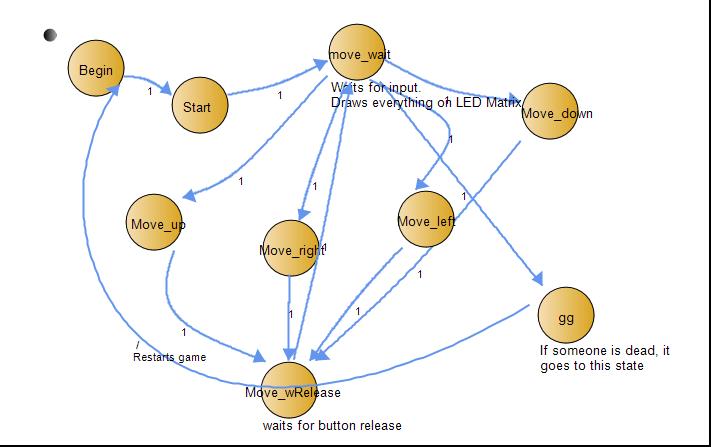
Technical design

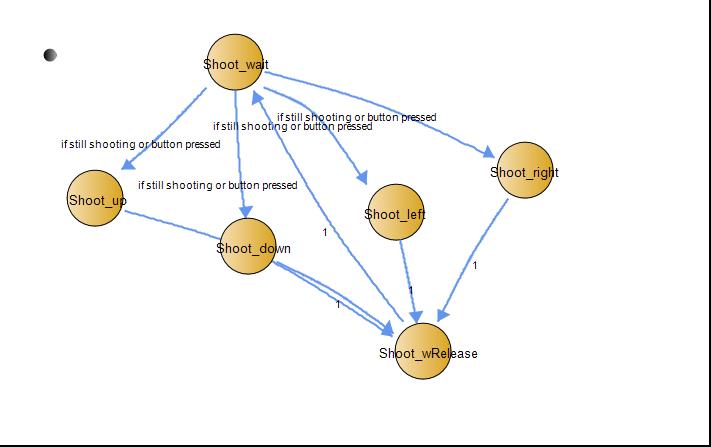
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PORTA | PORTB | PORTC | PORTD |
| 0 | LCD(out) | Shift Register(in) | Keypad(in) | LCD(out) |
| 1 | LCD(out) | Shift Register(in) | Keypad(in) | LCD(out) |
| 2 | Speaker(out) | Shift Register(in) | Keypad(in) | LCD(out) |
| 3 | Nothing | reset (in) | Keypad(in) | LCD(out) |
| 4 | Nothing | Nothing | Keypad(in) | LCD(out) |
| 5 | Nothing | Nothing | Keypad(in) | LCD(out) |
| 6 | LED Bar(out) | Nothing | Keypad(in) | LCD(out) |
| 7 | LED Bar(out) | Nothing | Keypad(in) | LCD(out) |

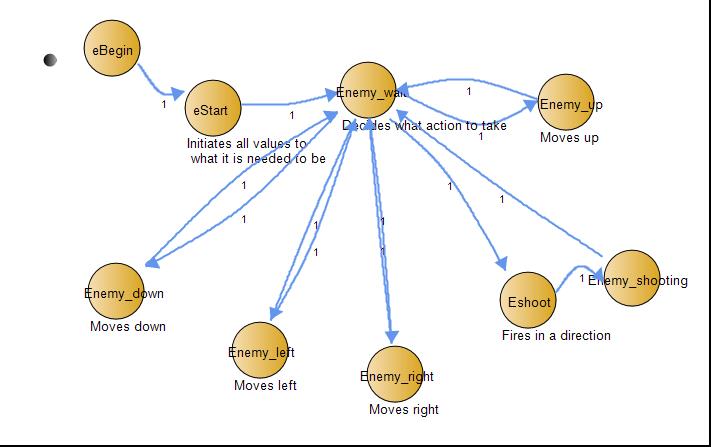


Breadboards wired up.

|  |  |
| --- | --- |
| SynchSM-name | Description |
| LI\_Tick | LCD initialization and display |
| LT\_Tick | LCD transaction |
| Moving | Game and user interface  This SM takes all the inputs and uses them appropriately |
| Shooting | Shooting projectiles for user |
| Enmove | Controls enemy movement and shooting |







Results and Performance Evaluation

Testing:

I spent a good amount of time testing and letting others play to ensure that this game would work fine.

This is what i tested.

* pressing buttons while game is over
* shooting between LED matrixes to ensure smoothness.
* shooting green walls
* shooting out of bounds
* can’t move through walls
* pressing multiple buttons
* making sure reset and on/off switch overrides everything
* making sure lcd displays the correct string
* outputting lights to make sure there is no dimming

Bugs:

1. Whenever I reset, There are some lingering lights on the LED matrix. I think this is because of me using 7 shift registers with one dataIN. I have tried clearing it after the game ends, but its just there.

2. I also have keypad 7 to be a reset button when the game ends, it would work most of the time, but sometimes it spawned the enemy at the wrong location or instantly ends the game. I couldn't fix this so I removed this feature.

3. There was a bug where I would die randomly from an enemy bullet, but I fixed that problem by double checking and changing the coordinates of the enemy bullet.

Conclusion

Final Remarks:

I spent a good amount of hours learning, wiring and testing how to use my shift registers and LED matrixes. I was pretty satisfied after figuring out how to use them and manipulate them using c code. My artificial intelligence in this game actually exceeded my expectations, because I originally wasn’t planning on having an enemy that can track you down no matter what location you are currently at. The method I used to display the map and having objects destroyed was to create multiple arrays and cases to change my output on my screen every time something was hit.

Youtube link: http://www.youtube.com/watch?v=6vncTc6r6tc