Tact-ical Fall

Custom Project Final Report

Summer 2018

Siwon Kim

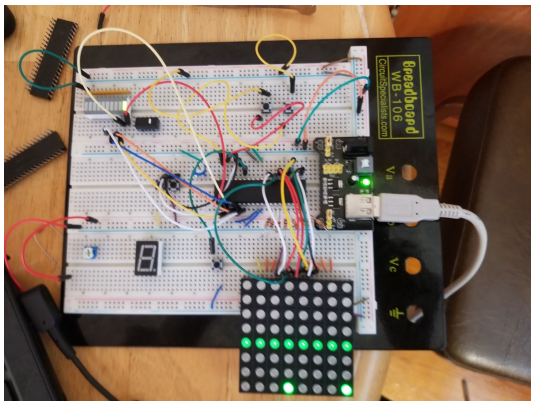
Table of Contents

Introduction

Tact-ical Fall requires users to avoid falling(random) objects in a row while avoiding the “catcher”. The game consists of 2 players each assigned a role. The first player(spawns in the middle) is assigned a role of avoiding the falling objects while avoiding the second player. The second player(the catcher, spawns at the right edge) attempts to catch first player.

The first player can die from getting hit by an object or getting catched by second player while the row of falling object is at the bottom. The first player can restart the game by pressing both left and right buttons at the same time (doesn’t have to be concurrently) when the game is over. The second player tries to catch the first player while the row is at the bottom. So that the second player wins. The second player moves at the same speed as the first player and the position of second player resets every time the row is at the bottom (second player is invincible from the objects). The score is accumulated when the first player survives when the object falls. The score is displayed in the LED bar in binary.

Insert a picture of your project

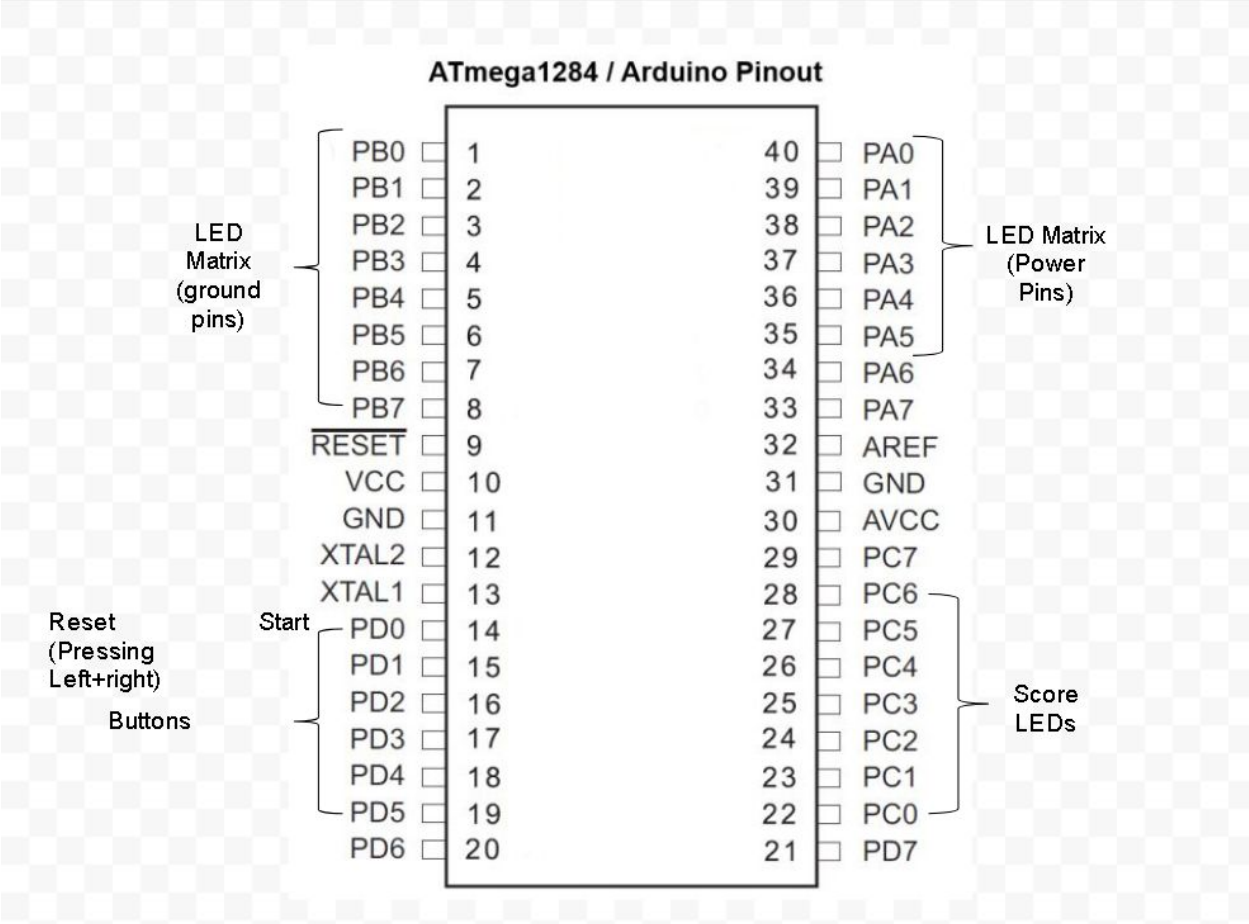


Hardware

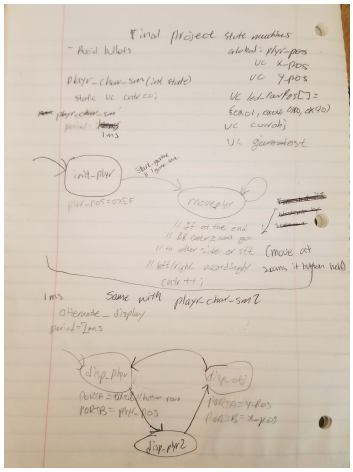
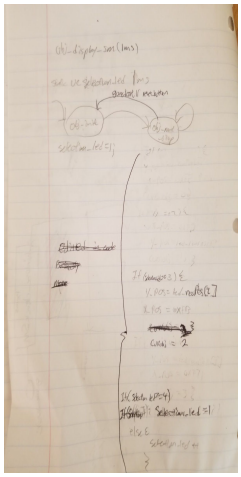
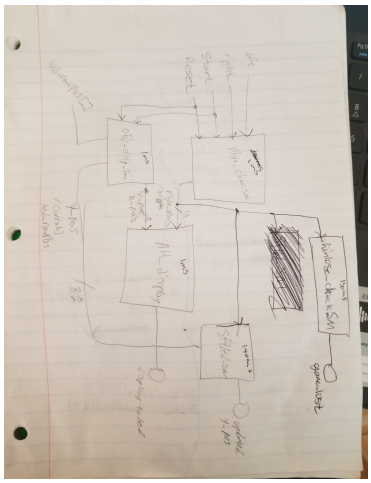
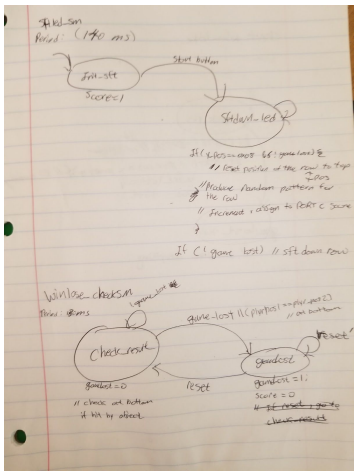
Parts List

- ATMega1284 microcontroller
- 10-LED bar
- Buttons
- 8x8 LED matrix

Pinout


ATmega1284 / Arduino Pinout
LED Matrix (ground pins)
LED Matrix (Power Pins)
Reset (Pressing Left+right)
Start
Buttons
Score LEDs
PB0 1
PB1 2
PB2 3
PB3 4
PB4 5
PB5 6
PB6 7
PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
PD0 14
PD1 15
PD2 16
PD3 17
PD4 18
PD5 19
PD6 20
40 PA0
39 PA1
38 PA2
37 PA3
36 PA4
35 PA5
34 PA6
33 PA7
32 AREF
31 GND
30 AVCC
29 PC7
28 PC6
27 PC5
26 PC4
25 PC3
24 PC2
23 PC1
22 PC0
21 PD7

Software

Used Atmel studio 7.0 to program

Complexities

Completed Complexities:

- Using LED Matrix
- Using game logic
- Implementing second player

Youtube link:
https://youtu.be/jpCufHL87AQ

Incomplete complexities:

- Shift register
- Joystick

Known Bugs and Shortcomings

- From the start of the game, sometimes the start button is not registered which I think it is caused by larger period of registering button than the button press.
- The game starts a bit late sometimes after pressing start. I think this is caused by arranging wrong variables in states for shifting objects.
- Only one life is given while I made 2 lives available. I think this is happening because the state where game checks if you won or lost is checking too fast for lives to decrease once at a time.

What I learned

- Expectations: I expected to learn more about using new components
- Planning: Thoroughly draw the state machines after planning out the product completely.
- Documentation: Organize the documents so that I can easily reference later.
- Timeline: 8/24-8/30
    - In future, plan out more carefully to make project better.

project code: https://github.com/skim258/cs120b-final-project