Week 14 - Project Progress 3rd Report
CIS-033, Spring 2024

Melissa Li
May 12, 2024

Table of Contents

Timeline & Current Status

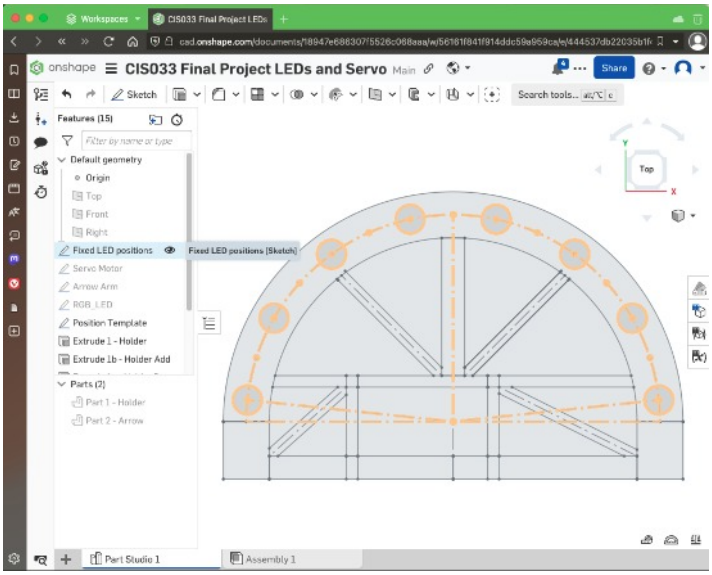
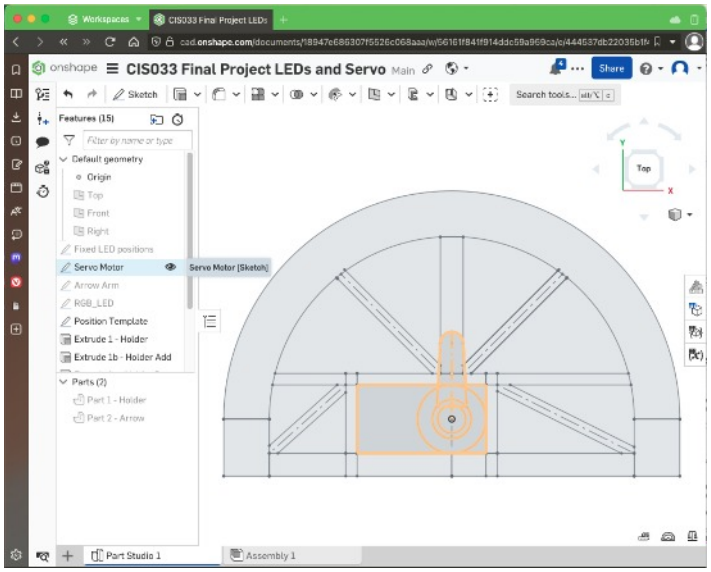
Week of	Done?	Planned Actions
April 22, 2024	x	• Project Timeline Plan *(this)*
	x	• Assemble components
	x	• Determine Arduino Pin Assignments for all components
	x	• Do Unit Tests, verify individual components / blocks
	x	• Prepare schematic with all connections
April 29, 2024	x	• Design & document Game Logic
	x	• Breadboard full circuit, with all components
	x	• Merge unit-test codes, verify components in single programs
	x	• Run startup tests, check all output components
May 6, 2024	x	• Mount RGB LED onto Servo-controlled part, and align fixed color LED in semi-circle around former
	x	• Code game logic
	x	• Test game mechanics
	(in-prog)	• Build structure to assemble all components suitable for game
		• Add simple sounds for game
May 13, 2024		• Last minute changes (if any)
		• Final polish

Work Done This Week

1. Mount for Servo & Fixed LEDs

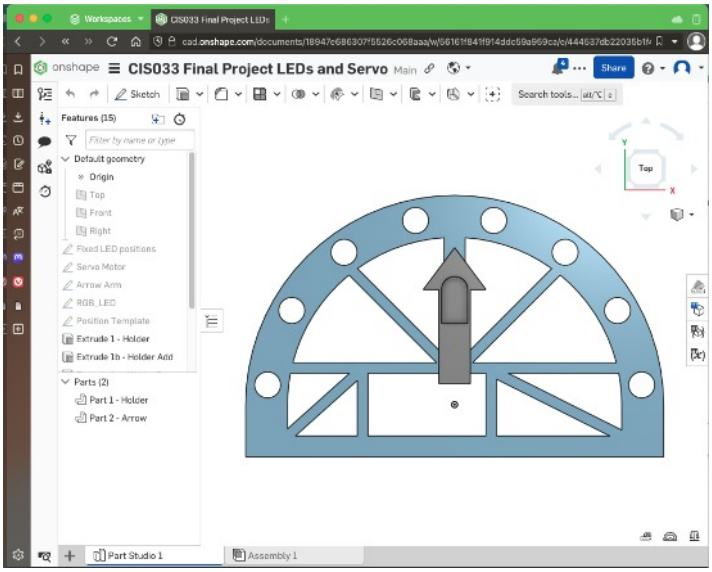
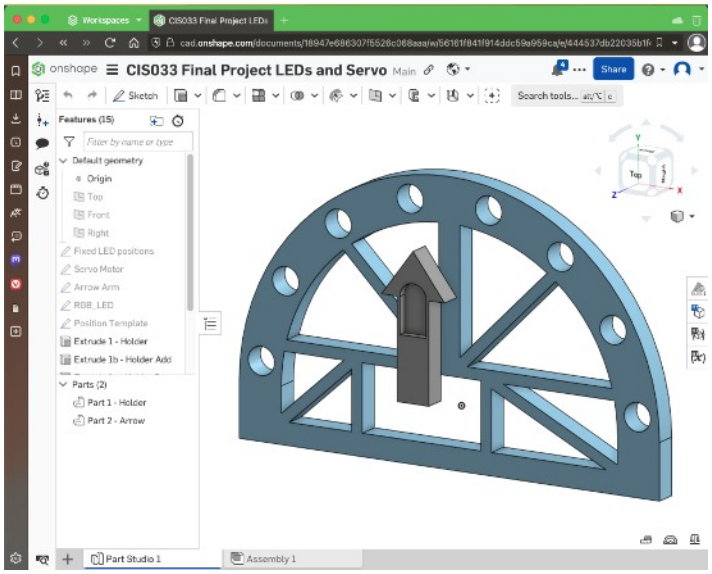
a) Sketches

Because component position plays a key part of the game mechanics, I used OnShape to draw up a mount to house the Servo-motor, in a central position around the Fixed LEDs. Doing this in CAD allows me to design a part with precise measurements.

Semi-circle Frame w/ Fixed LED positions	**Semi-circle Frame w/ Servo Motor placement (Motor rotation axis at Origin)**
	

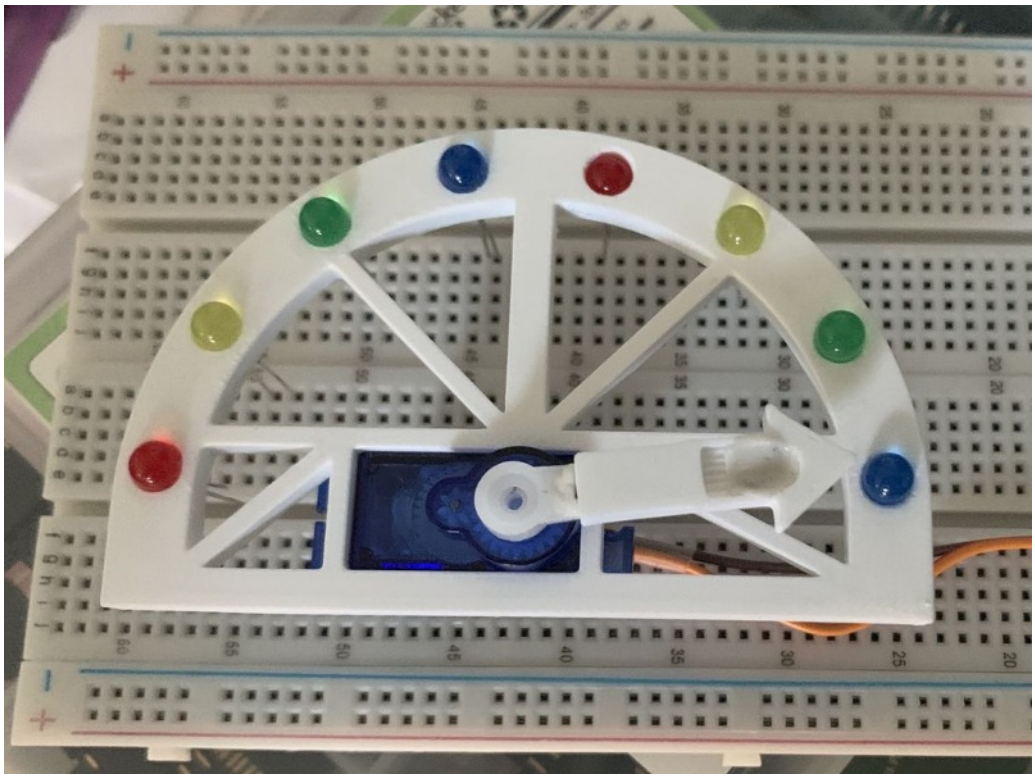
b) 3D Part in CAD

With sketches, I then extruded the necessary parts to form a 3D Model, of the Mounting Frame.

Top View	Angled View
	

c) 3D Part Printed

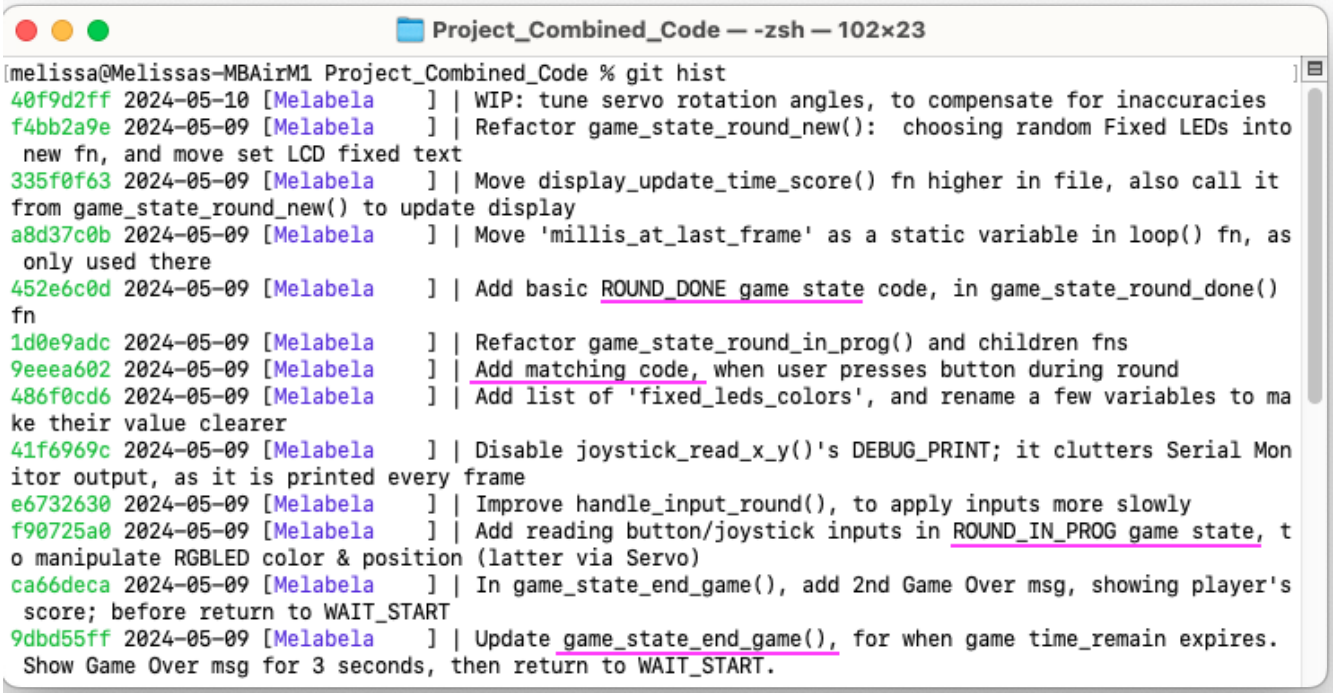
Then I used my partner's 3D Printer, to create the physical part, so I can mount the Fixed LEDs and Servo to it.



2. Programmed Game Logic

Partial Git history, showing some code changes this week.

Magenta Underline markup indicates commits involving game states / important logic.



3. Tested Game Mechanics

(No photos... forgot to take pictures with Prototype circuit. Dismantled to build final setup.)

But I can summarize the Game States:

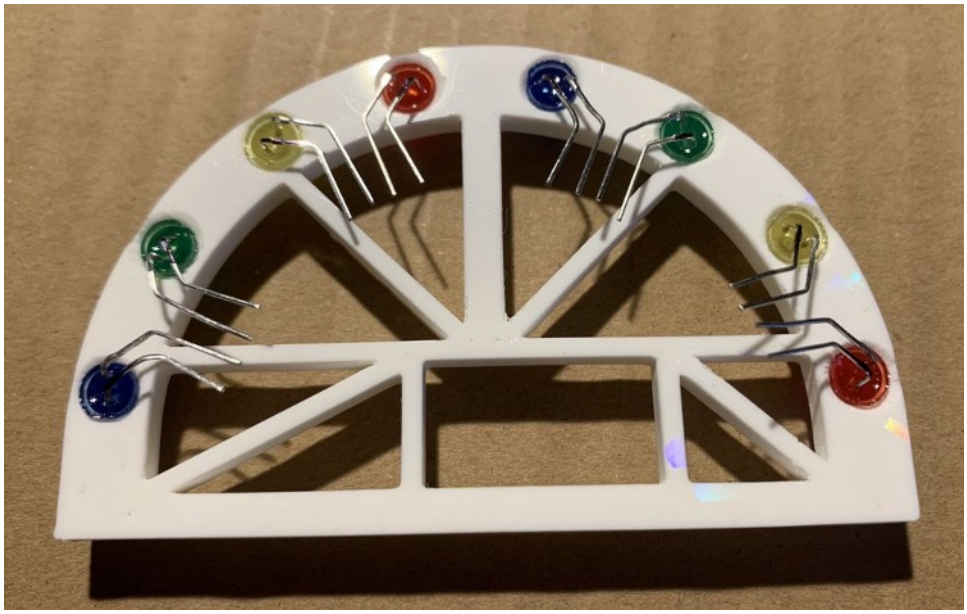
State	LCD Display	Actions
Power On	(init & clear screen)	• SETUP Inputs / Ouputs • GOTO "Wait Start" state
Wait Start	`"Press Button to Start Game."`	• Wait here. • ACCEPT Button Press input, to GOTO "Start Game" state
Start Game	`"Time left: <NN>.<n> Score: <M>"`	• Set "Time Left" to 30 seconds • Set "Score" to 0 • GOTO "Round New" state
Round New		• Randomly choose 1-3 Fixed LEDs to light, for user to match, in this round
Round In-Prog(ress)		• Count down "Time Left". If "Time Left" == 0, GOTO "End Game" state • ACCEPT Joystick X-axis input, to rotate Servo, to point RGB_LED to an adjacent

		Fixed LED • ACCEPT Joystick Y-axis input, to change RGB_LED to a different color • ACCEPT Button Press input, to check if RGB_LED color matches with Fixed LED's at that position. ◦ If Yes, then turn off that Fixed LED, and "Score" += 1
Round Done		• (Sound effects?) • GOTO "Round New" state, to reset LEDs to match, and continue game
End Game	" `Time Over. Game Over.`" ➔ for 3 seconds " `Game Over. Score: <N>.`" ➔ for another 3 seconds	After LCD Display messages, • GOTO "Wait Start" state

4. Assembling game components

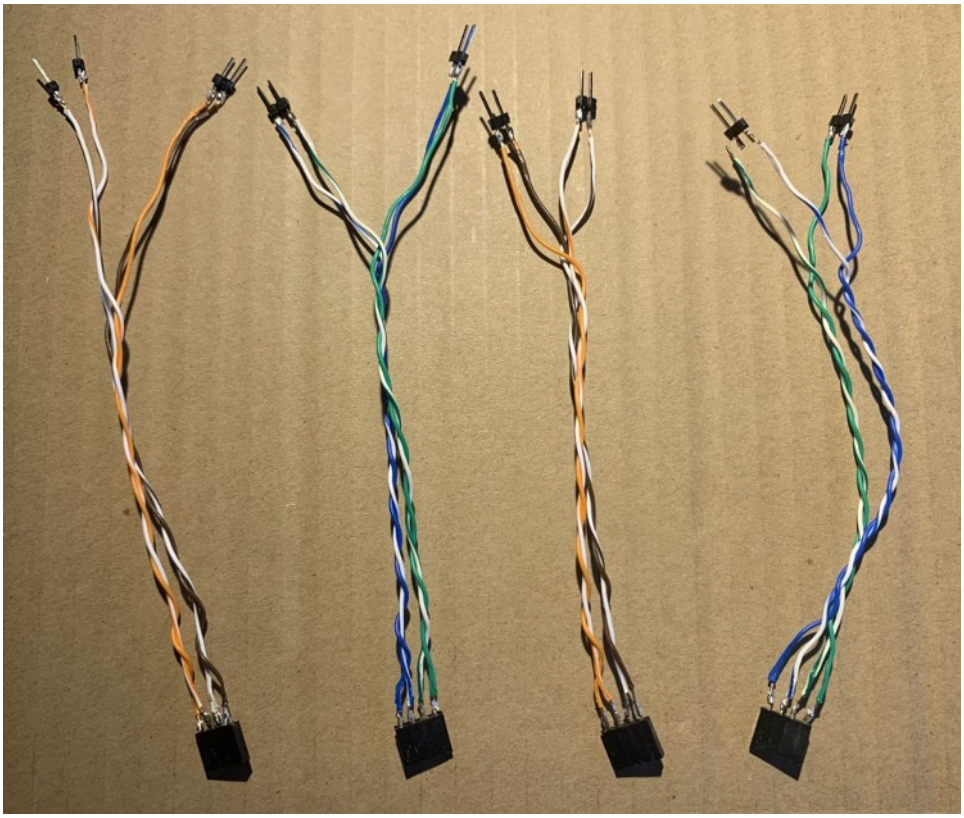
Now that Fixed LEDs are mounted on the frame in a semi-circle, need to rewire and reconnect them.

1. Mount the Fixed LEDs into the frame, and bend the leads on the underside into groups, 2 per LED.



2. Solder wires to jumper connectors. This is to avoid directly soldering to the LEDs, in case a mistake is made, so I can easily replace the LEDs if required.

- Used wires taken from inside an Ethernet cable, as they were already color-coded, to match the colors I was using (wires/LEDs -> brown/red, orange/yellow, green, and blue)



3. TODO: use above wire jumpers to reconnect Fixed LEDs, to components to ICs on the breadboard
    - 74HC595 Shift Register, and
    - 4116R-01-221 220 ohm Resistor Array