

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

Melissa Li
May 12, 2024

Table of Contents

Timeline & Current Status.....2

Work Done This Week.....3

 1. Mount for Servo & Fixed LEDs.....3

 2. Programmed Game Logic.....5

 3. Tested Game Mechanics.....5

 4. Assembling game components.....6

Timeline & Current Status

Week of	Done?	Planned Actions
April 22, 2024	x	<ul style="list-style-type: none"> Project Timeline Plan (<i>this</i>)
	x	<ul style="list-style-type: none"> Assemble components
	x	<ul style="list-style-type: none"> Determine Arduino Pin Assignments for all components
	x	<ul style="list-style-type: none"> Do Unit Tests, verify individual components / blocks
	x	<ul style="list-style-type: none"> Prepare schematic with all connections
April 29, 2024	x	<ul style="list-style-type: none"> Design & document Game Logic
	x	<ul style="list-style-type: none"> Breadboard full circuit, with all components
	x	<ul style="list-style-type: none"> Merge unit-test codes, verify components in single programs
	x	<ul style="list-style-type: none"> Run startup tests, check all output components
May 6, 2024	x	<ul style="list-style-type: none"> Mount RGB LED onto Servo-controlled part, and align fixed color LED in semi-circle around former
	x	<ul style="list-style-type: none"> Code game logic
	x	<ul style="list-style-type: none"> Test game mechanics
	(in-prog)	<ul style="list-style-type: none"> Build structure to assemble all components suitable for game
		<ul style="list-style-type: none"> Add simple sounds for game
May 13, 2024		<ul style="list-style-type: none"> Last minute changes (if any)
		<ul style="list-style-type: none"> Final polish

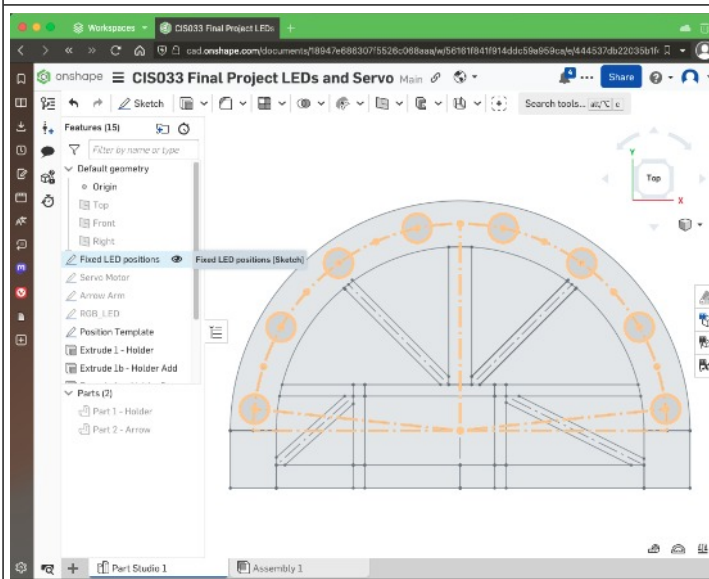
Work Done This Week

1. Mount for Servo & Fixed LEDs

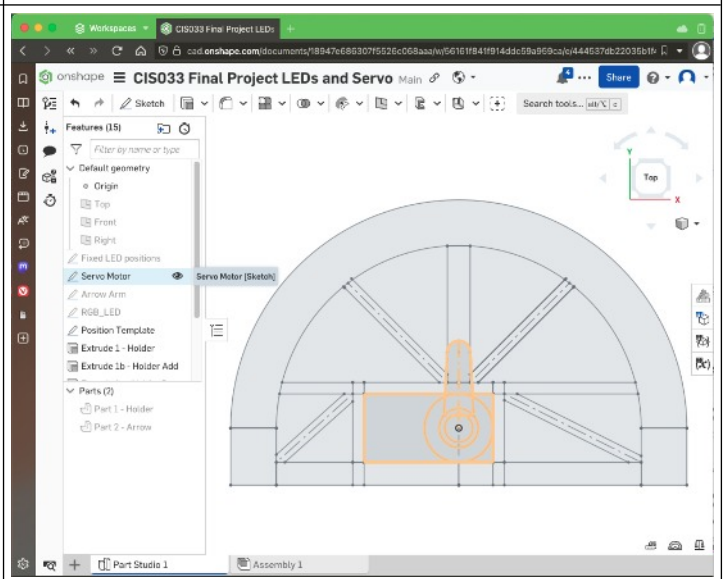
a) Sketches

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



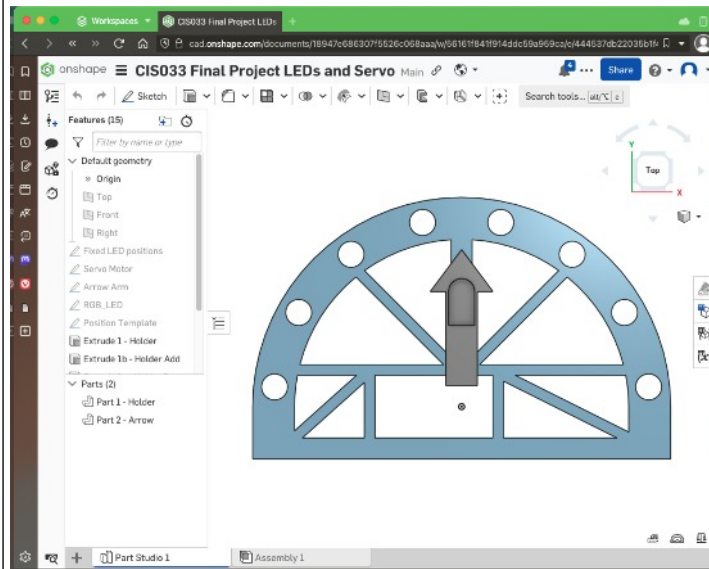
Semi-circle Frame w/ Servo Motor placement (Motor rotation axis at Origin)



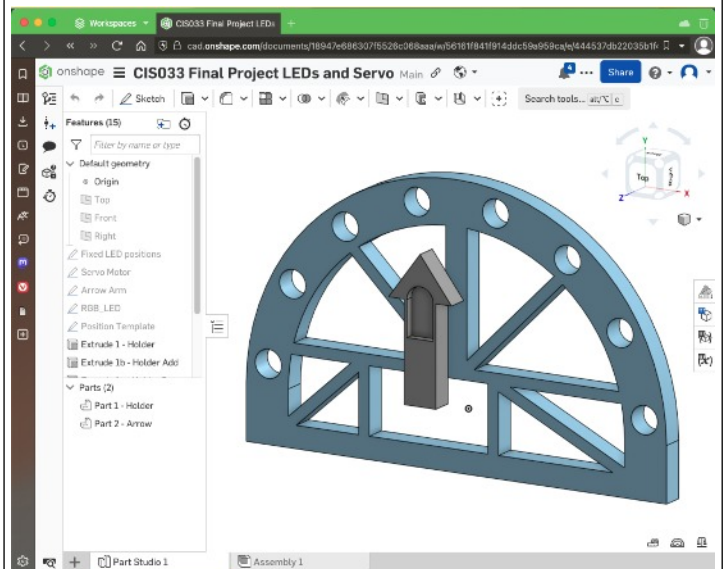
b) 3D Part in CAD

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

Top View

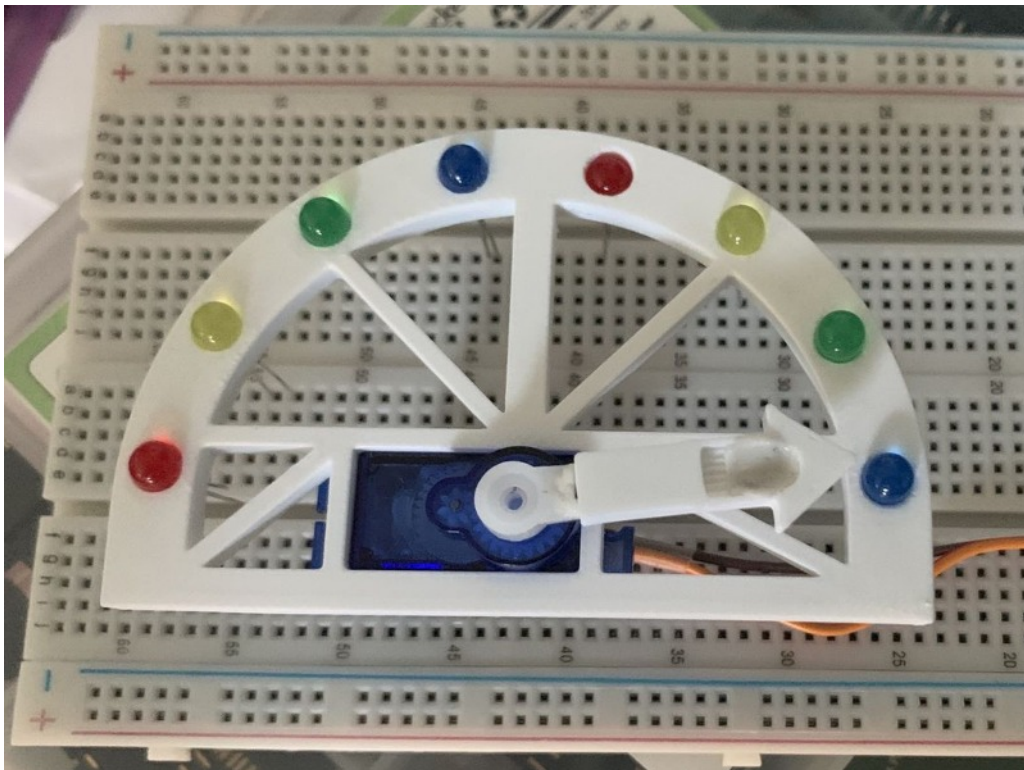


Angled View



c) 3D Part Printed

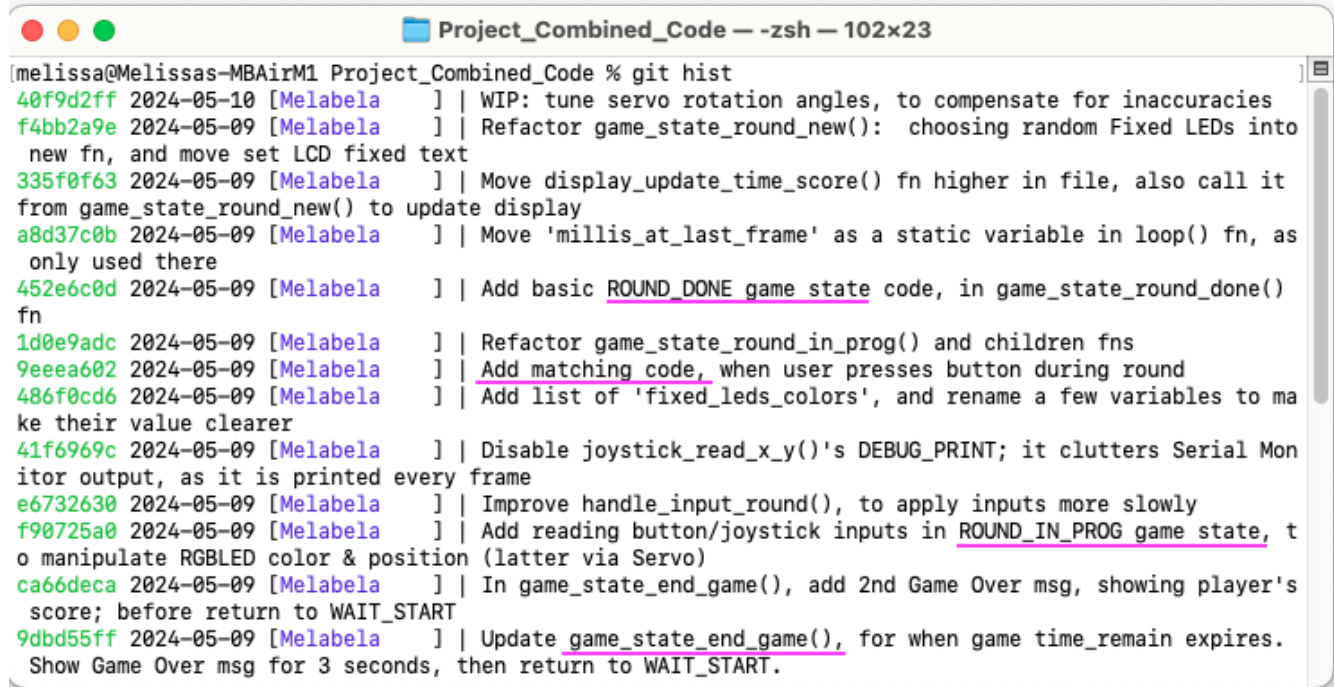
Then I used my partner's 3D Printer, to create the physical part, so I can mount the Fixed LEDs and Servo to it.



2. Programmed Game Logic

Partial Git history, showing some code changes this week.

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



```
melissa@Melissas-MBAirM1 Project_Combined_Code % git hist
40f9d2ff 2024-05-10 [Melabela] | WIP: tune servo rotation angles, to compensate for inaccuracies
f4bb2a9e 2024-05-09 [Melabela] | Refactor game_state_round_new(): choosing random Fixed LEDs into
new fn, and move set LCD fixed text
335f0f63 2024-05-09 [Melabela] | Move display_update_time_score() fn higher in file, also call it
from game_state_round_new() to update display
a8d37c0b 2024-05-09 [Melabela] | Move 'millis_at_last_frame' as a static variable in loop() fn, as
only used there
452e6c0d 2024-05-09 [Melabela] | Add basic ROUND_DONE game state code, in game_state_round_done()
fn
1d0e9adc 2024-05-09 [Melabela] | Refactor game_state_round_in_prog() and children fns
9eeea602 2024-05-09 [Melabela] | Add matching code, when user presses button during round
486f0cd6 2024-05-09 [Melabela] | Add list of 'fixed_leds_colors', and rename a few variables to ma
ke their value clearer
41f6969c 2024-05-09 [Melabela] | Disable joystick_read_x_y()'s DEBUG_PRINT; it clutters Serial Mon
itor output, as it is printed every frame
e6732630 2024-05-09 [Melabela] | Improve handle_input_round(), to apply inputs more slowly
f90725a0 2024-05-09 [Melabela] | Add reading button/joystick inputs in ROUND_IN_PROG game state, t
o manipulate RGBLED color & position (latter via Servo)
ca66deca 2024-05-09 [Melabela] | In game_state_end_game(), add 2nd Game Over msg, showing player's
score; before return to WAIT_START
9dbd55ff 2024-05-09 [Melabela] | Update game_state_end_game(), for when game time_remain expires.
Show Game Over msg for 3 seconds, then return to WAIT_START.
```

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:

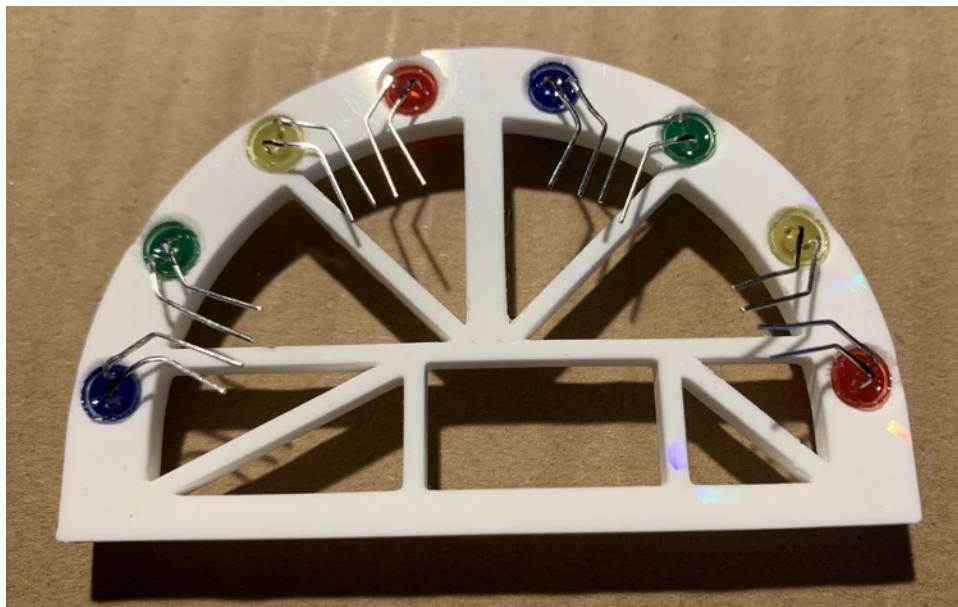
State	LCD Display	Actions
Power On	(init & clear screen)	<ul style="list-style-type: none">SETUP Inputs / OuputsGOTO "Wait Start" state
Wait Start	"Press Button to Start Game."	<ul style="list-style-type: none">Wait here.ACCEPT Button Press input, to GOTO "Start Game" state
Start Game	"Time left: <NN>.<n> Score: <M>"	<ul style="list-style-type: none">Set "Time Left" to 30 secondsSet "Score" to 0GOTO "Round New" state
Round New		<ul style="list-style-type: none">Randomly choose 1-3 Fixed LEDs to light, for user to match, in this round
Round In- Prog(ress)		<ul style="list-style-type: none">Count down "Time Left". If "Time Left" == 0, GOTO "End Game" stateACCEPT Joystick X-axis input, to rotate Servo, to point RGB_LED to an adjacent

		Fixed LED <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> ◦ If Yes, then turn off that Fixed LED, and "Score" += 1
Round Done		<ul style="list-style-type: none"> • (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, <ul style="list-style-type: none"> • GOTO "Wait Start" state

4. Assembling game components

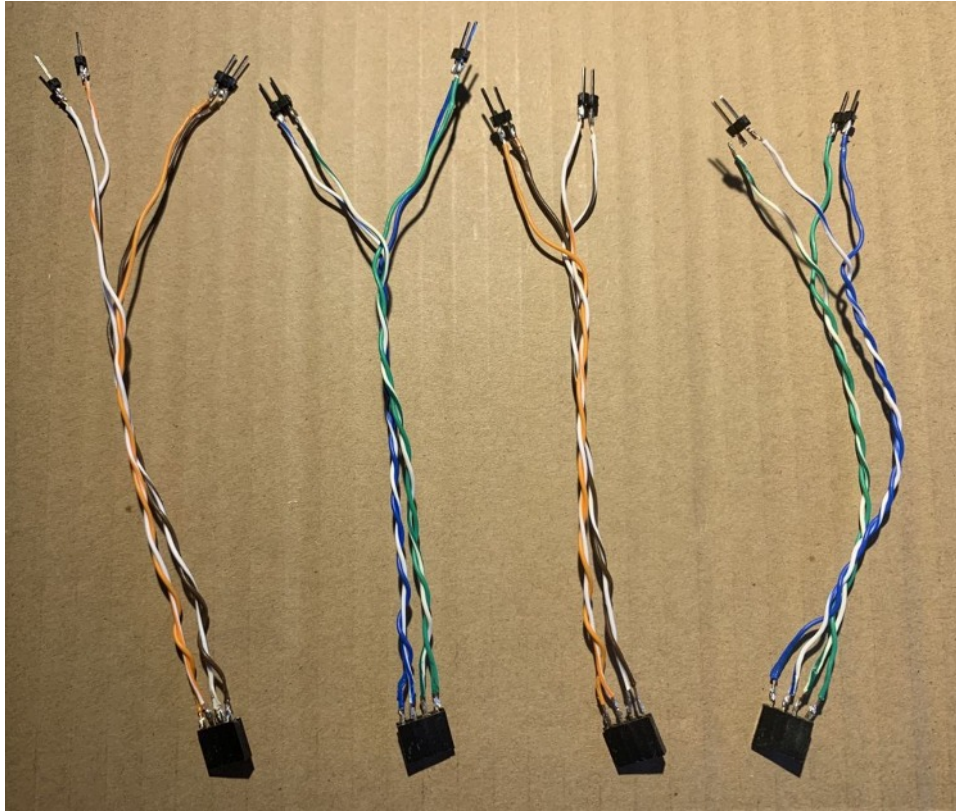
Now that Fixed LEDs are mounted on the frame in a semi-circle, need to rewire and reconnect them.

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