Project 3:

Design for Living Systems

Emma Murdock
DESN 22848

Interaction Design Media Motion and the Body

Table of Contents

1 Design Statement

This project is the culmination of explorations into construction of interactable devices and objects with conductive and non-conductive materials, p5.js and Makey Makey.

The result is a tactile and intuitive control board, joystick, and stylus linked to an on-screen canvas, allowing a user to draw with a variety colors, brush sizes, and opacities.

2 Week 9

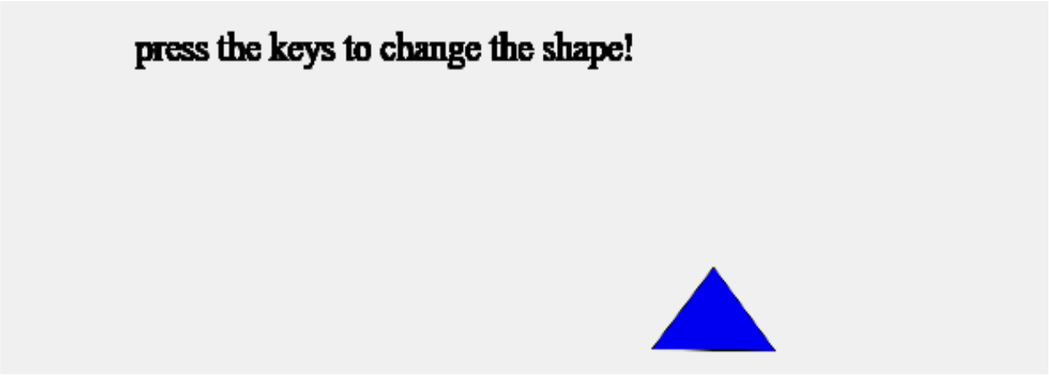
In week 9 I explored the various onscreen elements in p5.js I could control using key presses on my keyboard.

Two iterations from this week set up the work I did in my final: the color changing key presses, and the maze.

Color Changing Key Presses

In this interaction the user could press various keys in order to change the shape and color shown on screen.

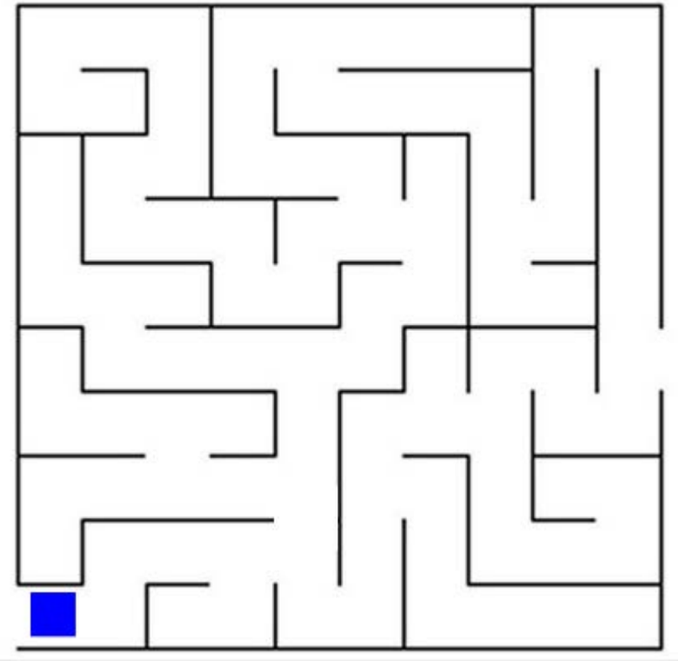
This inspired the color changing options in my final.



Maze

In this interaction the user could move a square through a maze using the arrow keys on the keyboard.

This inspired an iteration in Week 11 in which I linked the key presses to directions on a home-made joystick. This idea of using a joystick to control motion of an on-screen element is used in my final.



3 Week 10

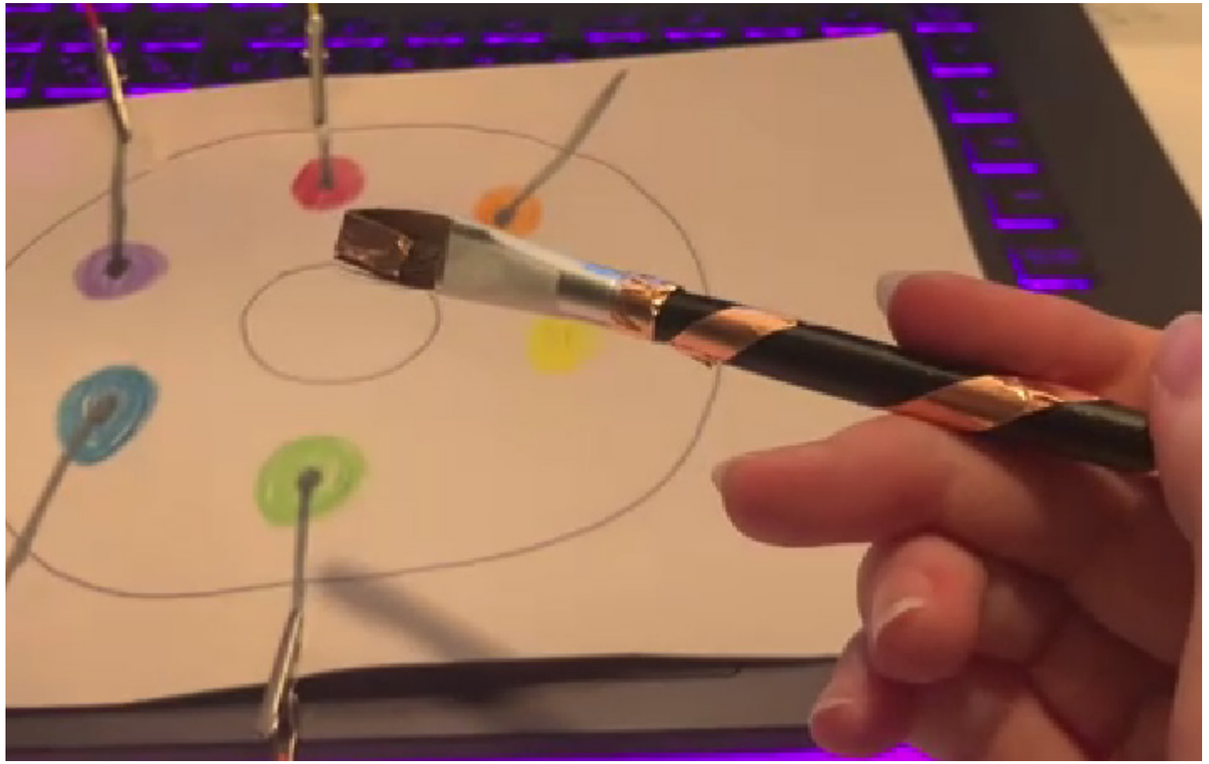
In week 10 I explored ways I could add a physical interaction beyond simple key presses to controllable elements in p5.js

The iteration I did this week which inspired my final is a colour palette hooked up to a conductive paintbrush.

Palette and Paint Brush

This iteration featured a 'palette' of colors, linked to Makey Makey keys through graphite lines, and a paintbrush wrapped in copper tape connected to the Makey Makey ground. This allowed the various colors on the palette to be touched with the brush in order to add the selected color an on-screen canvas.

This idea of a color palette hooked to traditional drawing tool, such as a paintbrush or pencil, was used in my final.



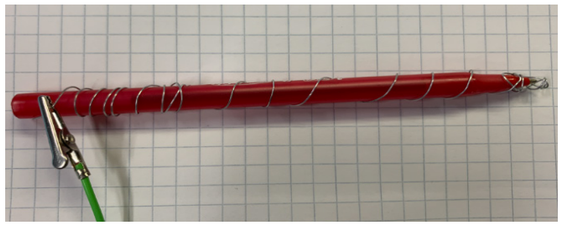
Week 11

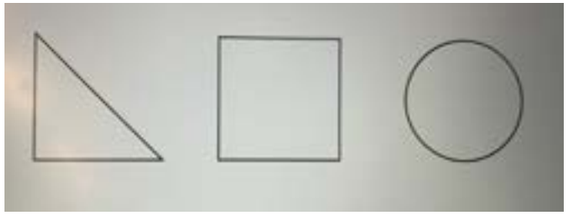
In week 11 I expanded on adding physical interactions by using a variety of 3D printed materials.

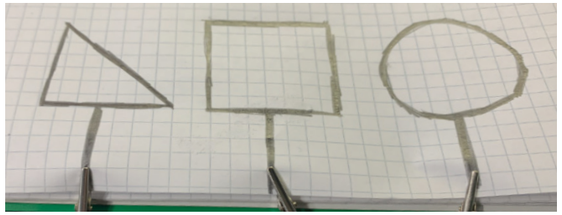
Two iterations which inspsired my final were the joystick and the conductive pen.

Conductive Pen.
In this interaction the could tap shapes on a piece of paper with a red pen, prompting the same shape on-screen to be colored red.

This, along with the color palette iteration from week 10, inspired the color changing and stylus elements of my final.



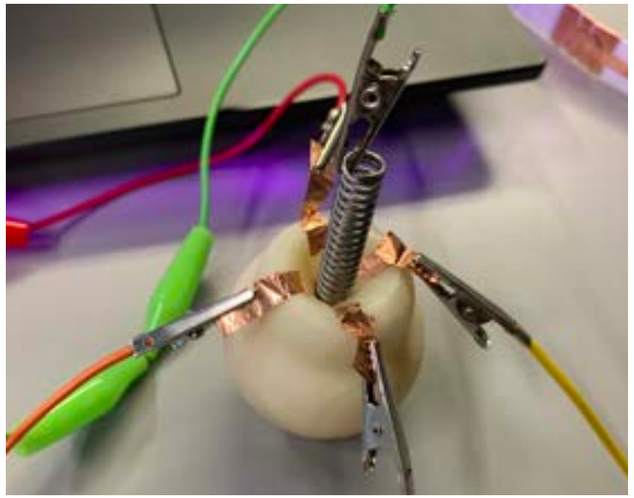




Joy Stick

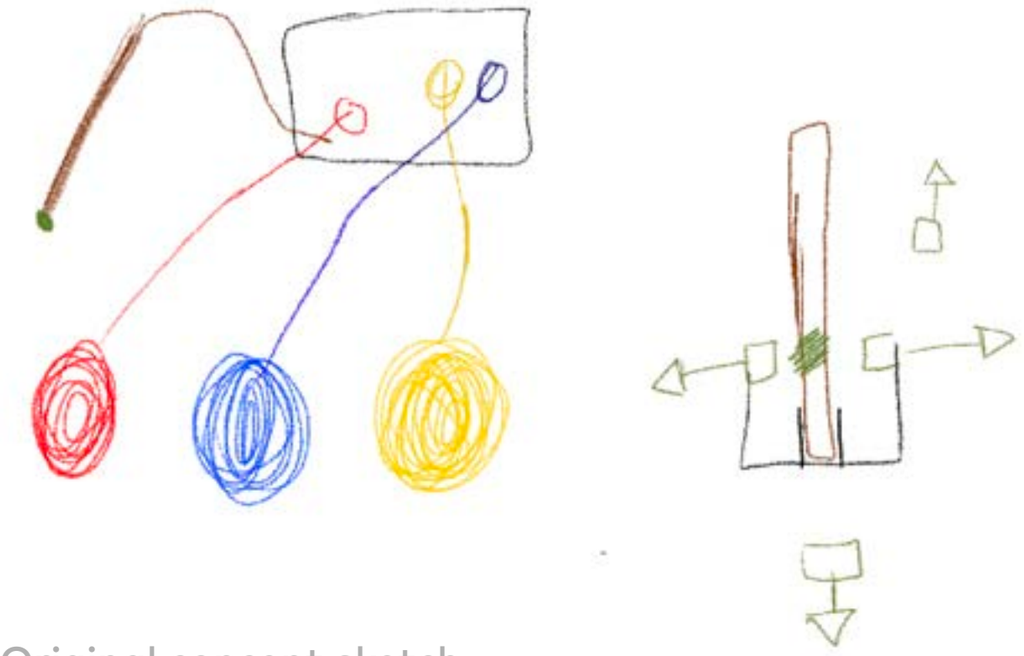
This iteration added a aspect to the Maze iteration from week 9. I constructed a joystick using an asbstract 3D printed shape and fitting a spring in the center (as is is conductive and allows movement) which could make contact with copper tape to the top, bottom, left, and right of it to trigger directional movement of the square from the maze iteration.

The design of this joy stick directly inspired the design of the joy stick I constructed in my final.



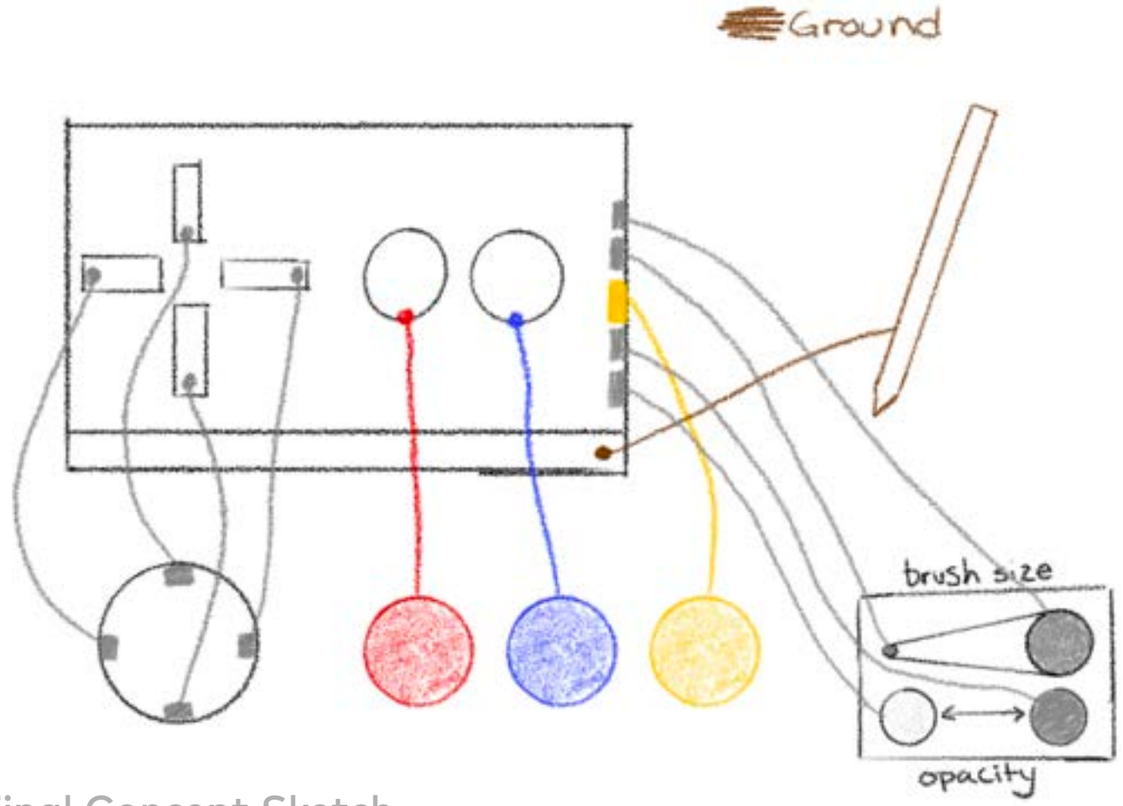
5 Concept

Combining various iterations from my weekly explorations, my concept was a physical control board that could be tapped with a stylus in order to change the color of an onscreen brush, which could then be moved to draw with by placing the stylus into an interface to turn itinto a joystick.



Original concept sketch

In expanding my idea further I decided to add additional inputs to control brush size and brish opacity, allowing the user more fine control.



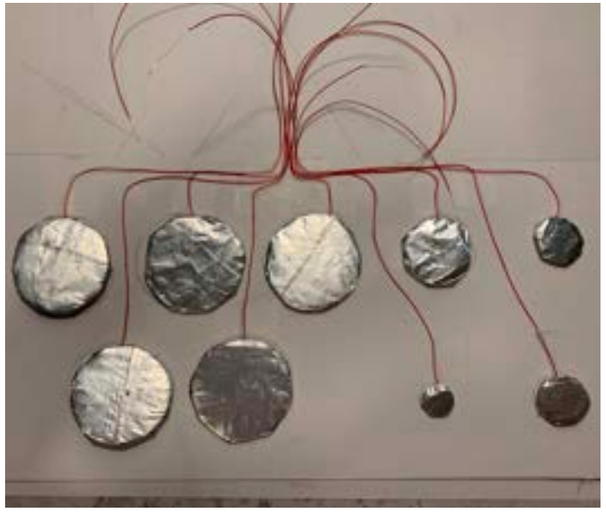
Final Concept Sketch

5 Construction

First I cut out the buttons for the color, size, and opacity and experiemnted with various layouts. I tried vertical and horizontal layouts for the size and opacity buttons, and a row of 3 and a row of 2 versus a square layout with a center button for the colors.

Next I traced my layout of buttons onto a piece of cardboard and cut out where the buttons were going to be in order to make a cover for the inside of the controls so it would look professional and clean.

Then I attatched pieces of wire to the back of the buttons in order to hook them up to the Makey Makey, and organized them neatly on the board so all the wired would converge in at the top in so could wrap them together for convience.



6 Construction

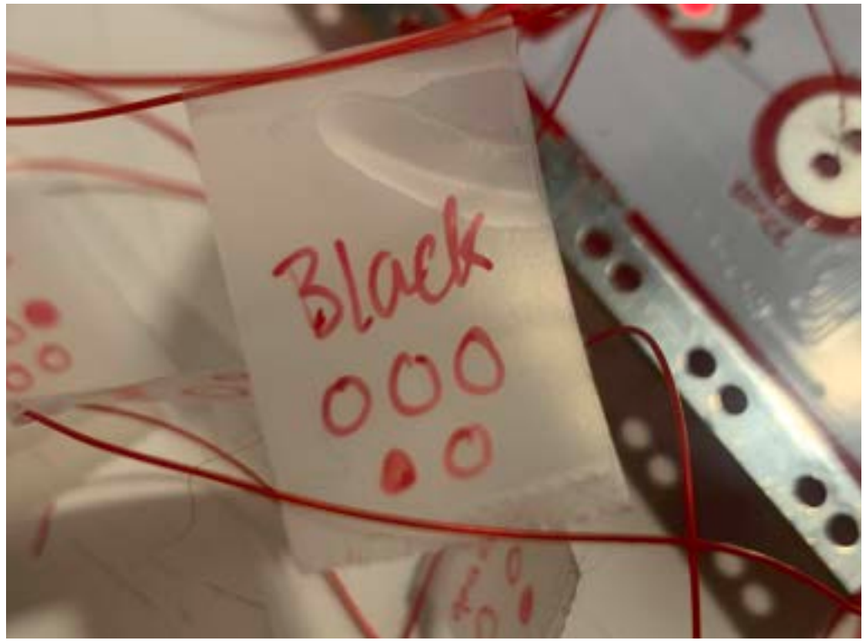
Next I extended the wires, wrapping the exposed wire in tape so multiple wires would not touch eachother and break electricity flow.

Then I labbeled each wire with what button it was attatched to on the board and what on-screen element it was going to control.

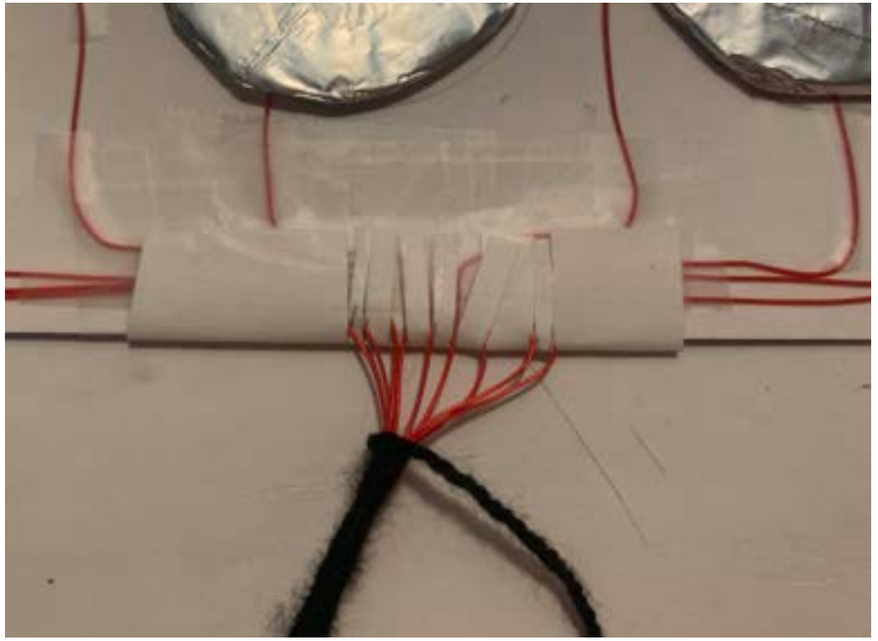
Then I wrapped all the wire together in order to make the board and accesories more convient to move. However I did learn in this process that I should have wrapped the wires last, as I did have to time-consumingly unwrap the wires multiple times to fix mistakes.

How I extended the wires, later wrapped in tape

Label on a wire



Wires wrapped together



7 Construction

Next I made the joy stick. Inspired my design in week 11, I made a base out of a small plastic bottle which I put a paper tube in the middle of so the stylus would stand upright.

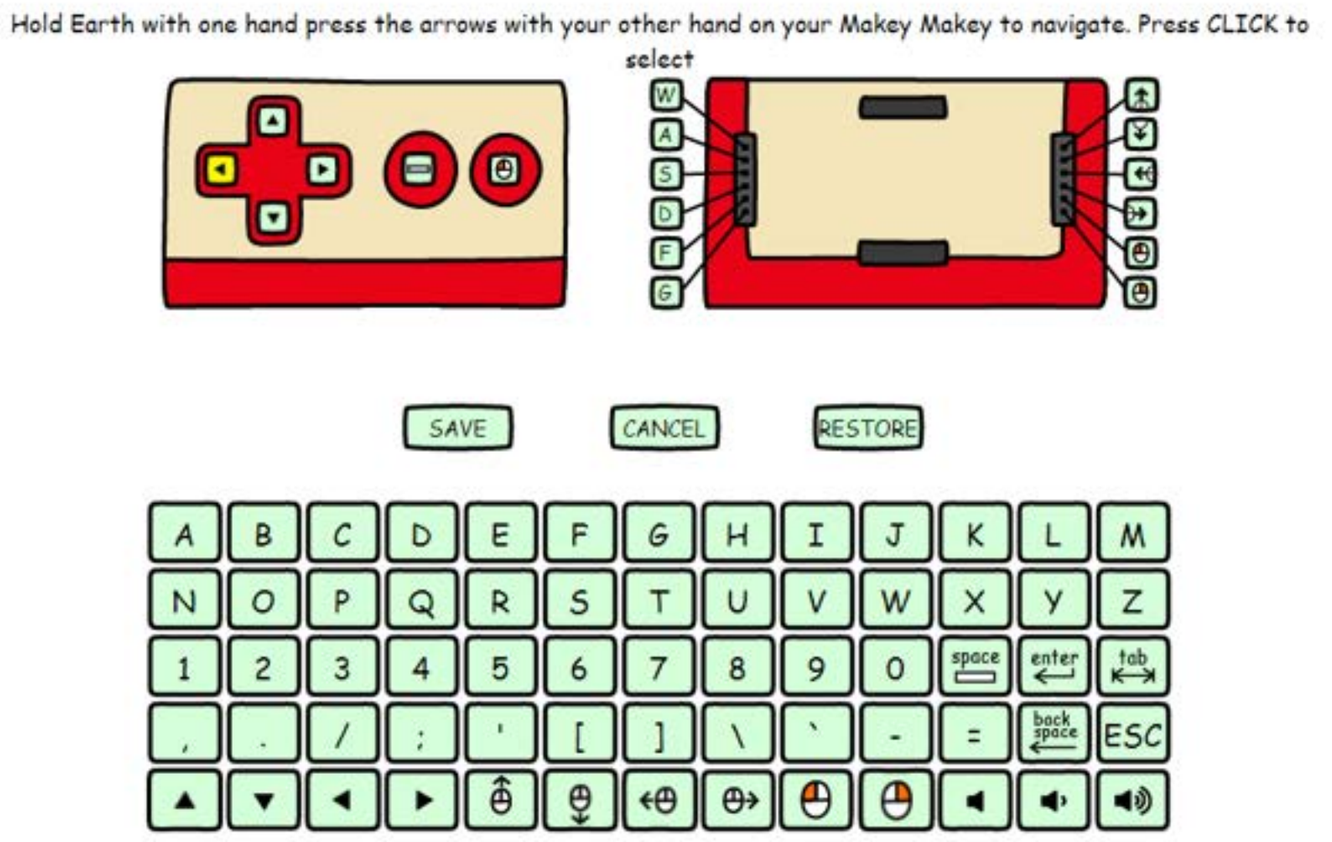
Then I put pieces of tinfoil around the edge of the tube and attached wires to those pieces of tinfoil in so that each piece could be hooked up to a key on the Makey Makey.

I later covered both the inside and outside of the tube in paper to make it look for clean and professional.

Extra Inputs and Code

One problem I ran into was the key presses that the inputs offered. I found that in order to have enough inputs, I would have to use the Right Click input, Left Click input, and either Space or Click. The problem with Right and Left click was that if I used them, it also triggered the Click option as they count as mouse clicks. I tried Changing the Click option to Space, but either my computer or p5.js has an issue with the space bar input, and every time Space was pressed the entire drawing would freeze and not move again.

In order to fix this I remapped my Makey Makey. Using the Remapping feature on the Makey Makey website I was able to change which keys the inputs on the back and front were tied to. I changed the Right Click, Left Click, and Space to letter keys, and the problem was solved.



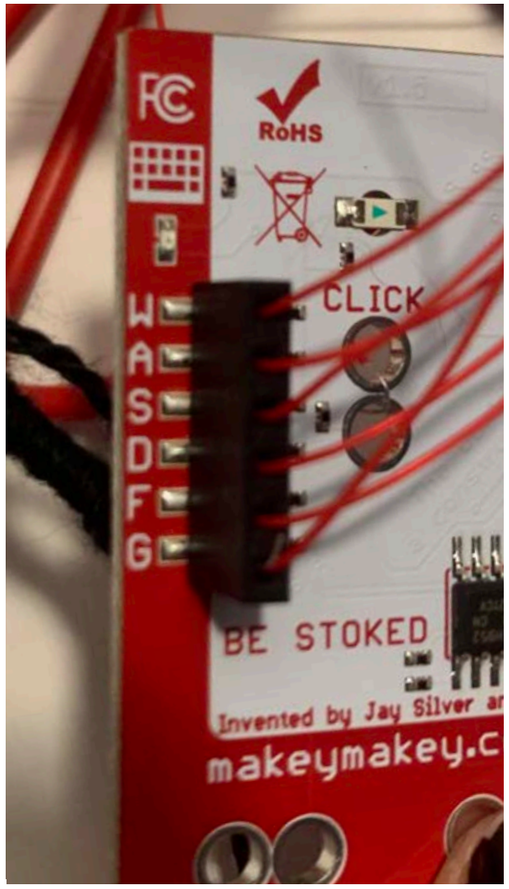
Makey Makey Remap page

Extra Inputs and Code

Because the makey makey only has 6 input options, and my control board has 13, I went online and learned about how to add more on makeymakey.com.

On the back of the Makey Makey there are extra input options that can be controlled with the W, A, S, D, F, and G keys, as well as left and right mouse clicks. Along with the normal inputs on the Makey Makey, this made 14 inputs total, so more than enough for what I needed.

Following a tutorial on the Makey Makey website I hooked up the extra inputs, wrote down which key was linked to each button, and began coding.



The code I used is fairly simple. I just had to add variables for the color, size, opacity, x coordinate, and y coordinate. However I did add some conditional statements in order to stop the user from moving the brush off the screen and trying to add above 100% opacity or below 0% opacity.

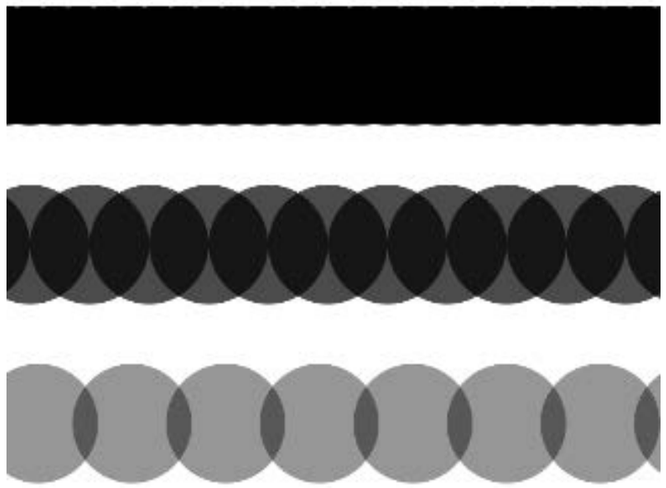
```
fill(r, g, b, opac);

noStroke();
ellipse(x, y, size, size);

keyDown();
```

```
} else if (key == 'f' && opac < 255) {
    opac += 75;
    move -= 20;

} else if (key == 'g' && opac > 0) {
    opac -= 75;
    move += 20;
```

```
if (keyIsDown(UP_ARROW) && y > 0) {
    y -= move;
    console.log('uparrow')

} else if (keyIsDown(LEFT_ARROW) && x > 0) {
    x -= move;

} else if (keyIsDown(DOWN_ARROW) && y < 710) {
    y += move;

} else if (keyIsDown(RIGHT_ARROW) && x < 1500) {
    x += move;
```

```
let r = 0;
let g = 0;
let b = 0;
let opac = 255;
let x = 750;
let y = 400;
let size = 70;
let move = 15;
```

10 Limitations



Top: Full opacity line
Middle: Mid opacity line
Bottom: Low opacity line

Drawing with p5.js really means drawing one shape, in this case a circle, one per frame at whatever the frame count may be. This means that the higher the frame count the more solid a line be, and the lower the frame count the more dotted it will look.

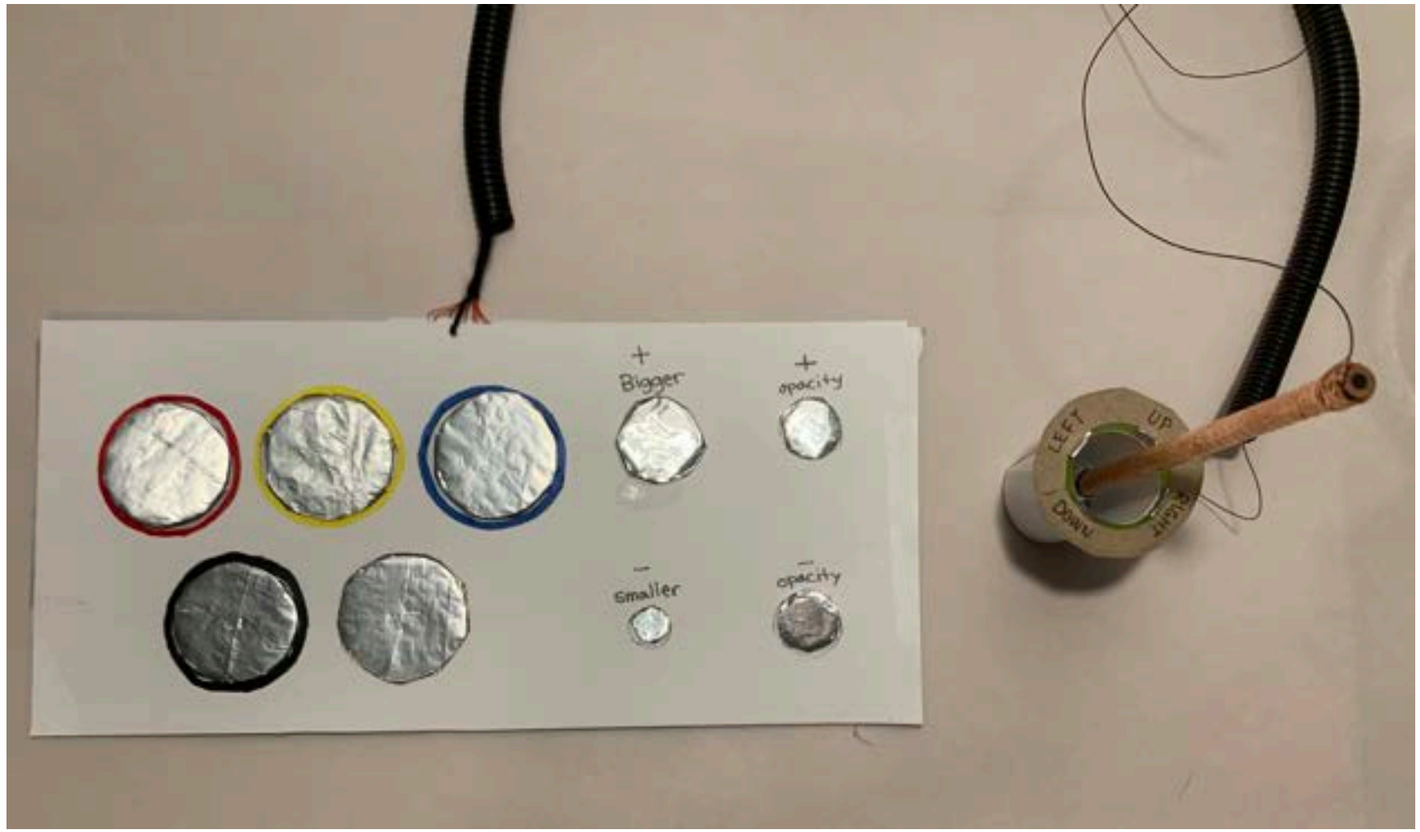
This creates an issue when lowering the opacity of a shape, as it then allows all any shapes from previous frames that overlap to be seen through the translucency of the current shape.

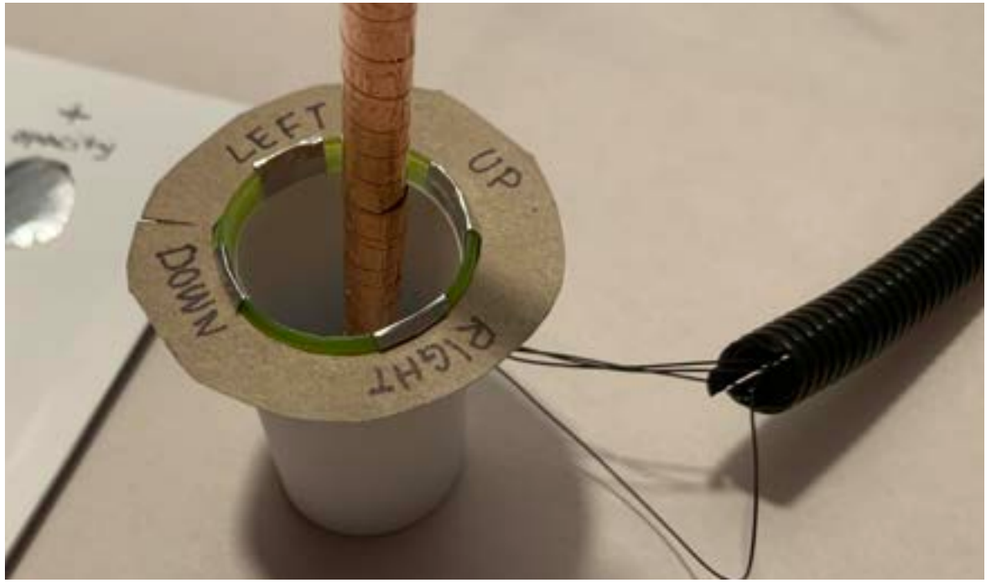
That issue, paired with the high frame count necessary to make a line seem solid, makes it impossible to see that the opacity is lowered because of how close together the shapes are. In order to remedy this in my code, I lowered the frame count as the opacity lowers, putting more space between shapes and allowing the opacity to be seen.

However, due to the lower frame count, shapes from previous frames can be seen, creating a sort of dotted line with uneven opacity. I could not find any way to fix this within the capabilities of p5.js.

I considered making the controls change the brightness of the color instead of the opacity in order to keep the line solid, however that would limit users to different shades of only red, blue, and yellow. Adding more colors is also not an option, as there are not enough inputs on the Makey Makey. I decided to stay with opacity option, as it allowed users the ability to ‘mix’ colors.

11 Final Design





12 Works Cited

Makey Makey. “Back of the Board Tutorial- More Inputs!” Joylabz Official Makey Makey Store, makeymakey.com/blogs/how-to-instructions/back-of-the-board-tutorial.

Makey Makey. “REMAP.” Apps.makeymakey.com, apps.makeymakey.com/remap/.

Ye, Quinquin, and Lauren Lee McCarthy. “P5.Js | Home.” P5js.org, p5js.org/.