Week 7: Drawing machines

We're going deeeeeeep down the rabbit hole.

The good news is that you pretty much have all the coding fundamentals you need for now to reach for the coding stars. 

The next couple of exercises will be related to extending this knowledge through various other inputs and allowing ourselves to run free with this idea. This will give you an idea of where you can take your sketches from Task 1.

Physical inputs

Perhaps we want a different image making tool, other than the keys on our keyboard? Maybe we want a sculpture or a drawing to control our sketch. Using a Makey Makey, you can already do all this. You just need a couple more steps:

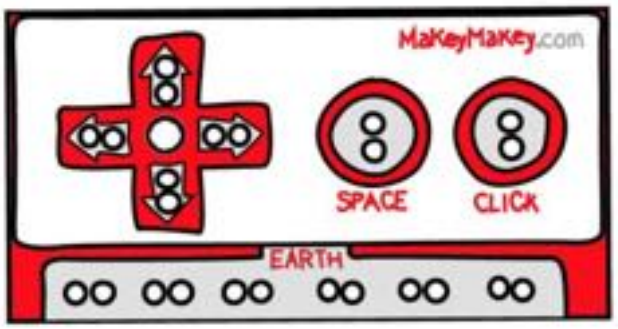
- Makey Makey board
- Alligator clips
- AND something conductive

<table>
<tr>
<td> 
Use play dough.</td>
<td> 
Use conductive adhesive copper tape.</td>
</tr>
<tr>
<td> 
Use a pencil and draw it!</td>
<td> 
Use jars of water!</td>
</tr>
<tr>
<td> 
Use fruit!</td>
<td> 
Use other people!</td>
</tr>
</table>

And so much more: https://labz.makeymakey.com/remixes

And the best part of this on is that we already know how to do all this! Makey Makey by default accepts the following keys:

UP, **DOWN**, **LEFT**, **RIGHT**, **SPACE** and **CLICK(LEFT)**. All you need to do is connect to the key you want, connect to the ground and you will have yourself a complete circuit.



If you look on the other side of the Makey Makey, you can also connect the keys **W, A, S, D, F** and **G**.

For those that are more adventurous, you can also get similar effects with an Arduino and a Capacity Touch Shield without having to “ground” yourself.

Machine A
Fruit salad machine

Dick Watkins, Australia b.1937 / *The Mooche* 1968 / Synthetic polymer paint (PVA) and oil on canvas / The James C Sourris, AM, Collection. Gift of James C Sourris, AM, through the Queensland Art Gallery | Gallery of Modern Art Foundation 2014. Donated through the Australian Government's Cultural Gifts Program / Collection: Queensland Art Gallery / © Richard John Watkins, 1968. Licensed by Viscopy, Sydney, 2015

Let's sketch out an idea for a drawing machine based on Dick Watkins' The Mooche (1968).

Step 1. Look! Really look.

When I look at this image, I feel that it's strengths are not just within the seemingly random shapes and odd colours - but in its ability to look both disheveled yet complete. I love the rough paint work that surrounds the edging of some of the shapes but I also like that others are completely hard-edged.

Step 2. Focus on a small goal and then build slowly.

For my first step, I want to be able to draw a shape when I press a key. Easy.
In addition to this I want to use the colours from the artwork.
In addition to this I don't want green shapes everywhere, I want to concentrate them to the left-hand side of the sketch.

```
color[] colorList = new color[5];

void setup(){
  size(900,900);
  background(255);
  colorMode(HSB, 360, 100, 100);
  colorList[0] = color(160,50,24);
  colorList[1] = color(223, 62, 24);
  colorList[2] = color(346, 76, 45);
  colorList[3] = color(35, 59, 45);
  colorList[4] = color(9, 39, 33);

  noStroke();
}

void draw(){
}

void keyPressed(){
  if (keyCode == UP){
    fill(colorList[0]);
    ellipse(random(200),random(height/2,
height),200,200);
  }
}
```

Step 3. Evaluate and experiment

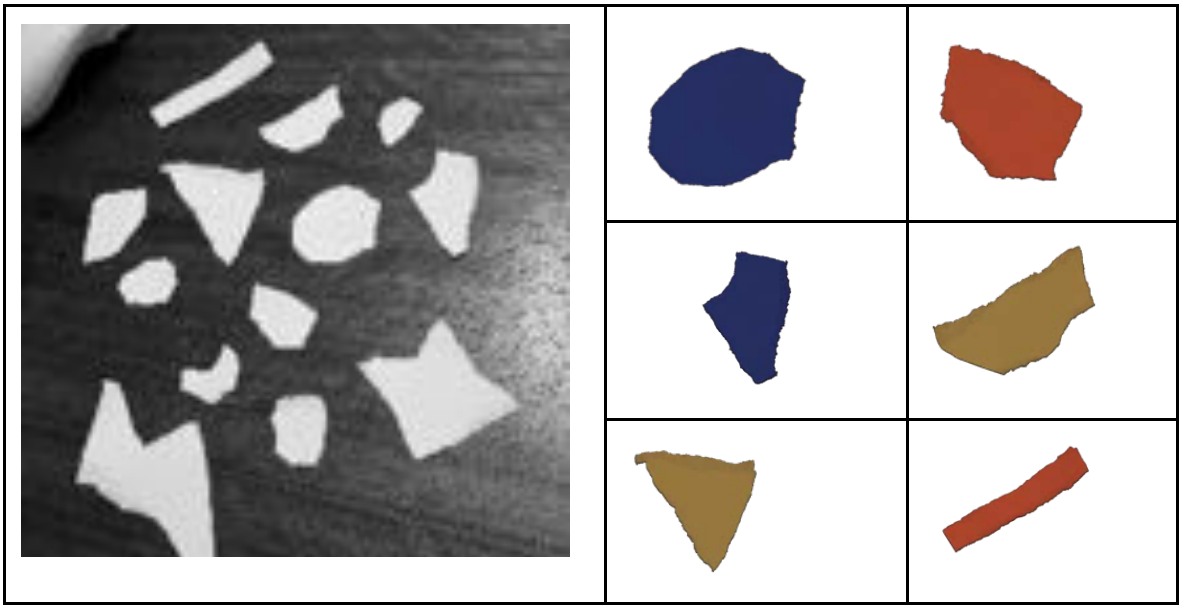
There are two things here that stand out to me:

- The canvas is way too white, nothing is that white in real life
- The circles are too perfect, they reek of “machine-made” - so how can we soften it?

So let’s add some random to the second shape that we will make ourselves and change the background colour.

```
color[] colorList = new color[5];

void setup(){
  size(900,900);
  background(255);
  colorMode(HSB, 360, 100, 100);
  colorList[0] = color(160,50,24);
  colorList[1] = color(223, 62, 24);
  colorList[2] = color(346, 76, 45);
  colorList[3] = color(35, 59, 45);
  colorList[4] = color(9, 39, 33);

  noStroke();
}

void draw(){
}

void keyPressed(){
  if (keyCode == UP){
    fill(colorList[0]);
    ellipse(random(-200,width/2),random(height),
    200,200);
  } else if (keyCode == DOWN){
    fill(colorList[1]);
    translate(random(-200,200), random(height));
    beginShape();
    vertex(random(350,370),random(150,170));
    vertex(random(400,500),random(80,100));
    vertex(random(450, 475), random(210, 240));
    vertex(random(400,450), random(290,310));
    endShape(CLOSE);
  }
}
```

Okay, still a bit gross but heading in the right direction. Next, I think I want to add a hands-on element to the interactive with a bit of paper ripping! Let’s see what that might look like?



So I make these tear out shapes and now I'm going to add them randomly to the screen.
When I hit the UP key, I'm going to print a small version of the random torn shape.
When I hit the DOWN key, I'm going to draw a larger torn shape.

```
color[] colorList = new color[5];
PImage[] shapes = new PImage[6];

void setup(){
  size(900,900);
  colorMode(HSB, 360, 100, 100);
  background(53,3,98);
  colorList[0] = color(160,50,24);
  colorList[1] = color(223, 62, 24);
  colorList[2] = color(346, 76, 45);
  colorList[3] = color(35, 59, 45);
  colorList[4] = color(9, 39, 33);
  shapes[0] = loadImage("assets/circle-blue.png");
  shapes[1] = loadImage("assets/circle-red.png");
  shapes[2] = loadImage("assets/line-blue.png");
  shapes[3] = loadImage("assets/circle-red-2.png");
  shapes[4] = loadImage("assets/quad-red.png");
  shapes[5] = loadImage("assets/triangle-gold.png");
  noStroke();
}

void draw(){
}

void keyPressed(){
  if (keyCode == UP){
    fill(colorList[int(random(5))]);
    ellipse(random(-200,width/2),random(height),200,200);
  } else if (keyCode == DOWN){
    fill(colorList[int(random(5))]);
    translate(random(100,500), random(height));
    beginShape();
    vertex(random(350,370),random(150,170));
    vertex(random(400,500),random(80,100));
    vertex(random(450, 475), random(210, 240));
    vertex(random(400,450), random(290,310));
    endShape(CLOSE);
  } else if (keyCode == RIGHT){
    scale(random(0.2, 0.5));
    image(shapes[int(random(6))], random(width), random(height));
  } else if (keyCode == LEFT){
    scale(random(0.75, 2));
    image(shapes[int(random(6))], random(width), random(height));
  }
}
```

Repeat Step 2 and 3 until you're happy:

- Step 2. Focus on a small goal and then build slowly.
- Step 3. Evaluate and experiment.

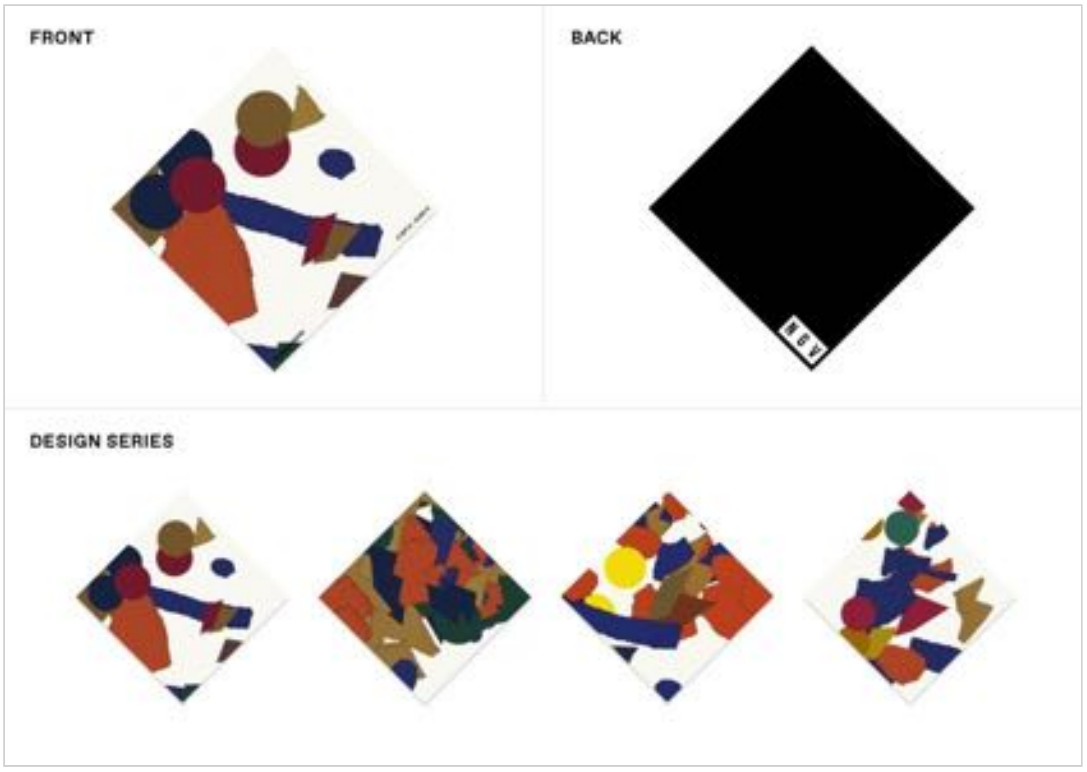
Here, I'm going to adjust some colours, refine some shapes and connect some conductives to demo.

```
color[] colorList = new color[5];
PImage[] shapes = new PImage[9];

void setup(){
  size(900,900);
  colorMode(HSB, 360, 100, 100);
  background(53,3,98);
  colorList[0] = color(160,53,38);
  colorList[1] = color(223, 62, 32);
  colorList[2] = color(346, 76, 55);
  colorList[3] = color(35, 85, 60);
  colorList[4] = color(45,79,66);
  shapes[0] = loadImage("assets/circle-blue.png");
  shapes[1] = loadImage("assets/circle-red-2.png");
  shapes[2] = loadImage("assets/circle-red.png");
  shapes[3] = loadImage("assets/jagger-gold.png");
  shapes[4] = loadImage("assets/line-blue.png");
  shapes[5] = loadImage("assets/line-red.png");
  shapes[6] = loadImage("assets/quad-blue-2.png");
  shapes[7] = loadImage("assets/quad-blue.png");
  shapes[8] = loadImage("assets/quad-red.png");
  noStroke();
}

void draw(){
}

void keyPressed(){
  if (keyCode == UP){
    fill(colorList[int(random(5))]);
    ellipse(random(-200,width/2),random(height),200,200);
  } else if (keyCode == DOWN){
    fill(colorList[int(random(5))]);
    scale(random(1.5,2));
    translate(random(-100,-100), random(height));
    beginShape();
    vertex(random(350,370),random(150,170));
    vertex(random(400,500),random(80,100));
    vertex(random(450, 475), random(210, 240));
    vertex(random(400,450), random(290,310));
    endShape(CLOSE);
  } else if (keyCode == RIGHT){
    scale(random(0.2, 0.5));
    image(shapes[int(random(9))], random(width), random(height));
  } else if (keyCode == LEFT){
    scale(random(0.75, 1.5));
    image(shapes[int(random(9))], random(width), random(height));
  }
}
```

And then the final puzzle piece, the poster design.

technecolour

Or perhaps we want to create a series of placemats for the NGV Design Store.



Or maybe some live wallpaper! Every time someone uploads a new image, it is tiled to this projection at the entrance of the exhibition. The latest tile is the centrepiece and the line of sight from where you create the tile to the projection is clear.

technecolour

Machine B
Spiral Machine

Col Jordan
Merrylands, New South Wales, Australia | born 1935
Daedalus - series 6 1968
paintings, synthetic polymer paint on canvas
164.0 h x 170.0 w x 4.0 d cm
framed (overall) 1660 h x 1670 w x 44 d mm

Step 1. Look! Really look.
The colours of this artwork are intensely striking compared to the other works in the exhibition. I love the Col Jordan has been able to create a dynamic rhythm in the work through both his use of colour and different angled and sized shapes.

Step 2. Focus on a small goal and then build slowly.
I want sample a random selection of colours directly from this work and reinterpret them into my own shapes and rhythms.

So how do we sample colours from an image?!
`pixels[]`

In Processing, when you load an image, you also have access to all its' pixels in a huge `Array`. Just like the colour palettes we've been making, pixels provides an `Array` of `colors` from an image. Let's give it a go.

```
PImage originalImage;

void setup(){
  background(255);
  size(1200,120);
  originalImage = loadImage("assets/Daedalus.png");
  originalImage.loadPixels();
}

void draw(){
}
```

Easy… but where are all our pixels?!

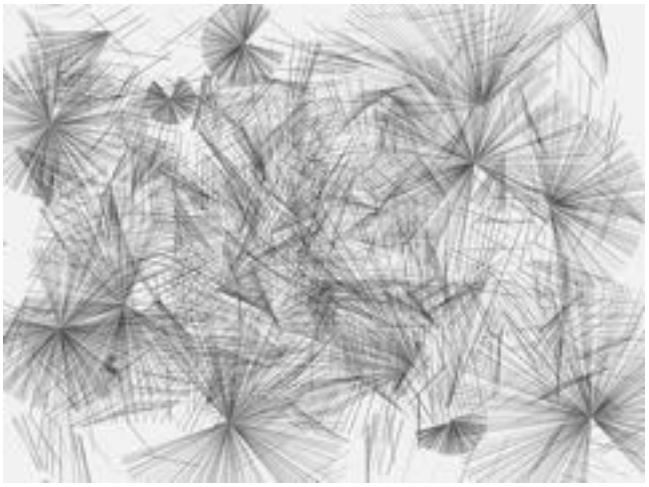
Well, we've only "loaded" the pixels but done nothing with them. The next step is to do something! I want to make a new `Array` of colours that has 10 colours and I want to put 10 random colours from the `pixels[] Array` into this new `Array`.

```
PImage originalImage;
//Create a new Array of 10 colours
color[] col = new color[10];

void setup(){
  background(255);
  size(1200,120);
  originalImage = loadImage("assets/Daedalus.png");
  originalImage.loadPixels();

  //pixels[] Array has (width x height) amount of pixels in image
  int dimension = originalImage.width * originalImage.height;
  //Pick a colour and add it to col[]. Run loop 10 times.
  for (int c = 0; c < col.length; c++){
    col[c] = originalImage.pixels[int(random(dimension))];
  }
}
```

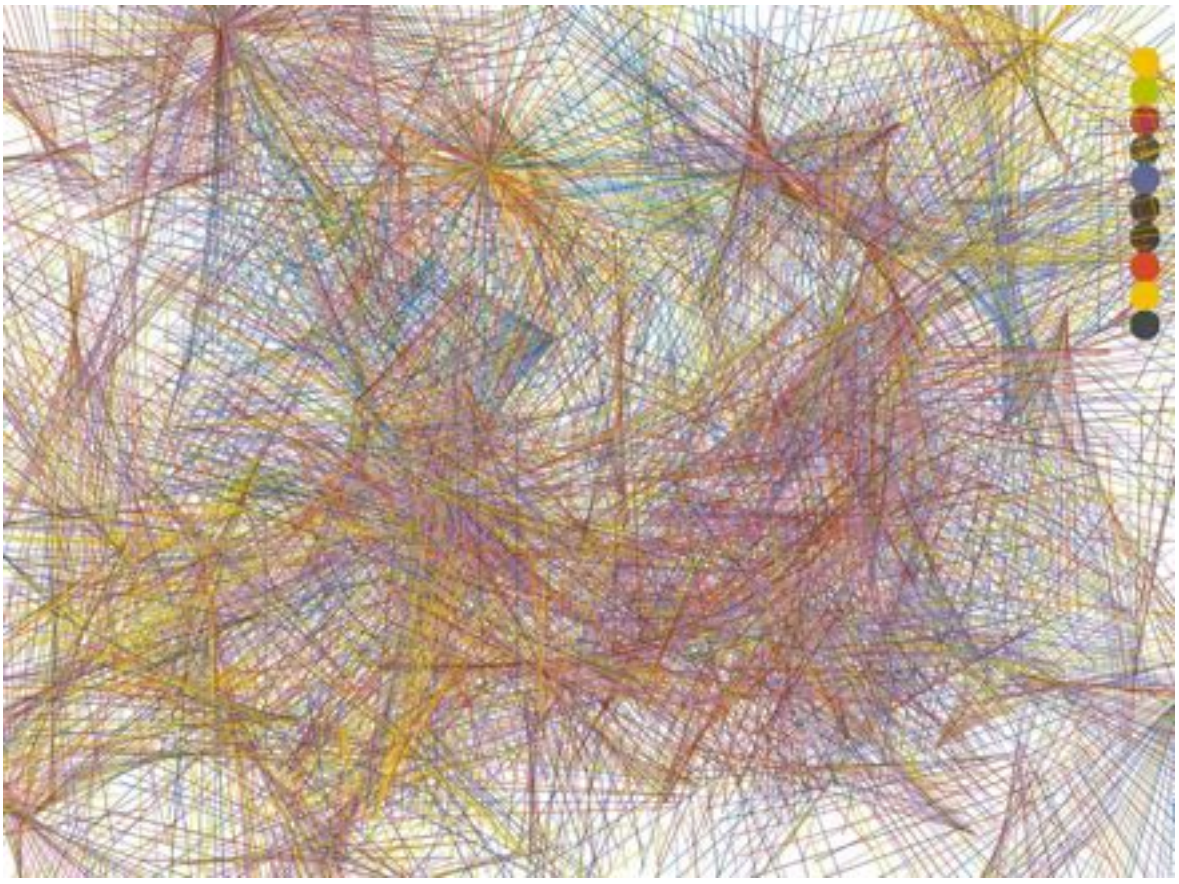
So now we have a 10 colours, I want to draw 10 squares and everytime I press a key, I want a new set of colours.

```
PImage originalImage;
//Create a new Array of 10 colours
color[] col = new color[10];

void setup(){
  background(255);
  size(1200,120);
  originalImage = loadImage("assets/Daedalus.png");
  originalImage.loadPixels();

  //pixels[] Array has (width x height) amount of pixels in image
  int dimension = originalImage.width * originalImage.height;
  //Pick a colour and add it to col[]. Run loop 10 times.
  for (int c = 0; c < col.length; c++){
    col[c] = originalImage.pixels[int(random(dimension))];
  }
}

void draw(){
  for (int i = 0; i < 10; i++){
    noStroke();
    fill(col[i]);
    rect(i*120,0,120,120);
  }
}

void keyPressed(){
  int dimension = originalImage.width * originalImage.height;
  for (int c = 0; c < 10; c++){
    col[c] = originalImage.pixels[int(random(dimension))];
  }
}
```

Step 3. Evaluate and experiment
I knew I wanted to use these random colours from my image as a colour palette, but I didn't know what exactly I wanted to create. And then I saw this sketch example of a swirling, spiral drawing machine.



```
/*
Generative Design: Visualise, Program and Create with Processing
P.2.3.1: Drawing with animated brushes (pg 236)

Modified to take the colour palette from an image 900 x 890 =
*/

float angle = 0;
float lineLength = 200;
float angleSpeed = 2;

void setup(){
  background(#f2f2f2);
  size(1200,900);
}

void draw(){
  if (mousePressed){
    pushMatrix();
    strokeWeight(1);
    noFill();
    stroke(random(255));
    translate(mouseX, mouseY);
    rotate(radians(angle));
    line(0,0,lineLength,0);
    popMatrix();
  }

  angle -= angleSpeed;
}

void mousePressed(){
  lineLength = random(400);
}
```



```
/*
Generative Design: Visualise, Program and Create with Processing
P.2.3.1: Drawing with animated brushes (pg 236)

Modified to take the colour palette from an image 900 x 890 =
*/

PImage originalImage;

color[] col = new color[10];
float angle = 0;
float lineLength = 200;
float angleSpeed = 2;

void setup(){
  background(255);
  size(1200,900);
  originalImage = loadImage("assets/Daedalus.png");
  originalImage.loadPixels();

  int dimension = originalImage.width * originalImage.height;
  for (int c = 0; c < col.length; c++){
    col[c] = originalImage.pixels[int(random(dimension))];
  }
}

void draw(){
  if (mousePressed){
    pushMatrix();
    strokeWeight(1);
```

technecolour

```
    noFill();
    stroke(col[int(random(col.length))]);
    translate(mouseX, mouseY);
    rotate(radians(angle));
    line(0,0,lineLength,0);
    popMatrix();
  }

  angle -= angleSpeed;
  for (int i = 0; i < col.length; i++){
    noStroke();
    fill(col[i]);
    ellipse(width-30, i*30 + 60, 30,30);
  }
}

void mousePressed(){
  lineLength = random(70,400);
}

void keyPressed(){
  if (keyCode == ENTER){
    int dimension = originalImage.width * originalImage.height;
    for (int c = 0; c < col.length; c++){
      col[c] = originalImage.pixels[int(random(dimension))];
    }
  }
}
```

Step 2. Focus on a small goal and then build slowly.
Step 3. Evaluate and experiment

I really liked the effect but instead of lines, I wanted rectangles like Col Jordan with different coloured fills and different widths.

Try to replace the lines with rectangles, see what madness you can come up with!

Step 2. Focus on a small goal and then build slowly.
Step 3. Evaluate and experiment

In this case I thought this pattern could make an incredible fabric or perhaps even a fabric AND a projection so I started looking for opportunities within the gallery.

Perfect. I would create bodysuits with my pattern on a stage projected with a similar pattern controlled by music.

I like how this is looking so maybe now I would expand upon this to demonstrate the sketch I use to draw the generative fabric and the sketch I use in the background controlled by music.
But now it's your turn.

technecolour

Week 7 Exercise

Take an artwork you haven't explored yet from The Field Revisited or related exhibitions and apply the three steps: Look, Focus and Evaluate.

Create a demo of an idea you have in the real world.

Reach out to examples, functions or methods you haven't tried before to see if they might spark new ideas.

Useful links

TUTORIALS

Daniel Shiffman Youtube Channel
If you haven't already discovered the magic that is… well, you're welcome. This guy is the greatest Processing teacher you will ever have. He will get you out of the darkest corner and show you that shiny shiny light.

Generative Design Book
This book has *plenty* of great examples that are easy to follow along.

LIBRARIES

Processing Libraries https://processing.org/reference/libraries/
Extend Processing beyond graphics and images into audio, video, and communication with other devices.

p5.js https://p5js.org/
Processing is a programming language based on Java. Javascript is another programming language that runs in your browser. p5.js is a library in Javascript based on Processing.

Confused yet? Don't worry, you only really need this for the next class so we can go through it together. However, here's a guide to port your Processing sketch to p5.

And luckily for us, there's a p5.js option in openProcessing. Daniel Shiffman also has a bunch of videos specific to learning p5.js!

ELECTRONICS

Makey Makey https://makeymakey.com/
The Make-a-Play-Dough-Keyboard I was talking about

Arduino https://www.arduino.cc/
Like Makey Makey, this is a board that you can customise - except this doesn't have to be just a keyboard. You can connect a sensors to test for all sorts of things.

Adafruit https://learn.adafruit.com/
A great resource for learning fun, relatively straight forward creative electronics projects. Also a good resource if you're looking to sew electronics.

ELECTRONICS - WHERE TO BUY

Pakronics https://www.pakronics.com.au/
Melbourne-based online electronics store - they have most things and do a speedy delivery to most places in Melbourne.

Little Bird Electronics https://www.littlebirdelectronics.com.au/
Sydney-based online electronics store - these guys have the best range and prices including a large selection of bits you'll find on Adafruit.