Agenda 10/13

- Open House Updates
- Smooth and organic motions:
  - Perlin Noise (More Array examples)
  - Matter.js

Random

let myNum = random(5,20)

Gives you a random number between 5 & 20 every time this line is executed

Every time a number is selected it is independent from the pervious number. While is this good in some cases, it is not always the best.

Let see this example: https://editor.p5js.org/scotchANDsolder/sketches/zz55GPMjM

The motion of the circle is too random! Does not look that interesting.

Perlin Noise

A better way of getting random numbers that are **smooth***

* - Perlin noise will give numbers that are dependent on the previous number

A much better way of moving circles:
https://editor.p5js.org/scotchANDsolder/sketches/xjmKGtfYa

Notice how the motion is more organic

Perlin Noise vs Random

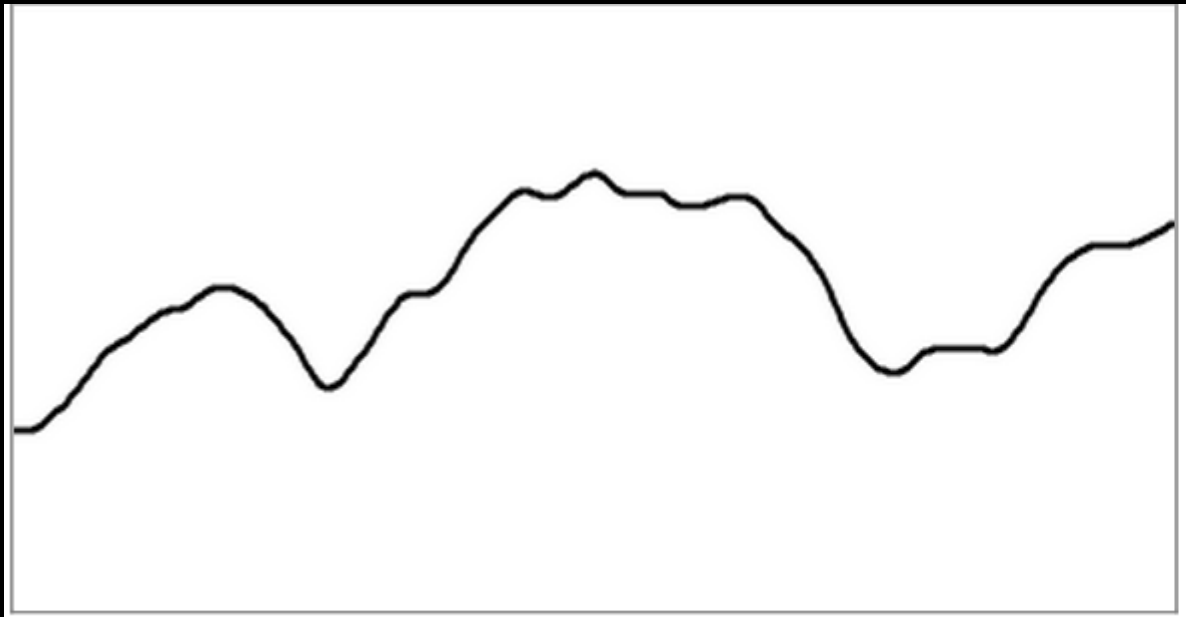

Figure I.5: Noise

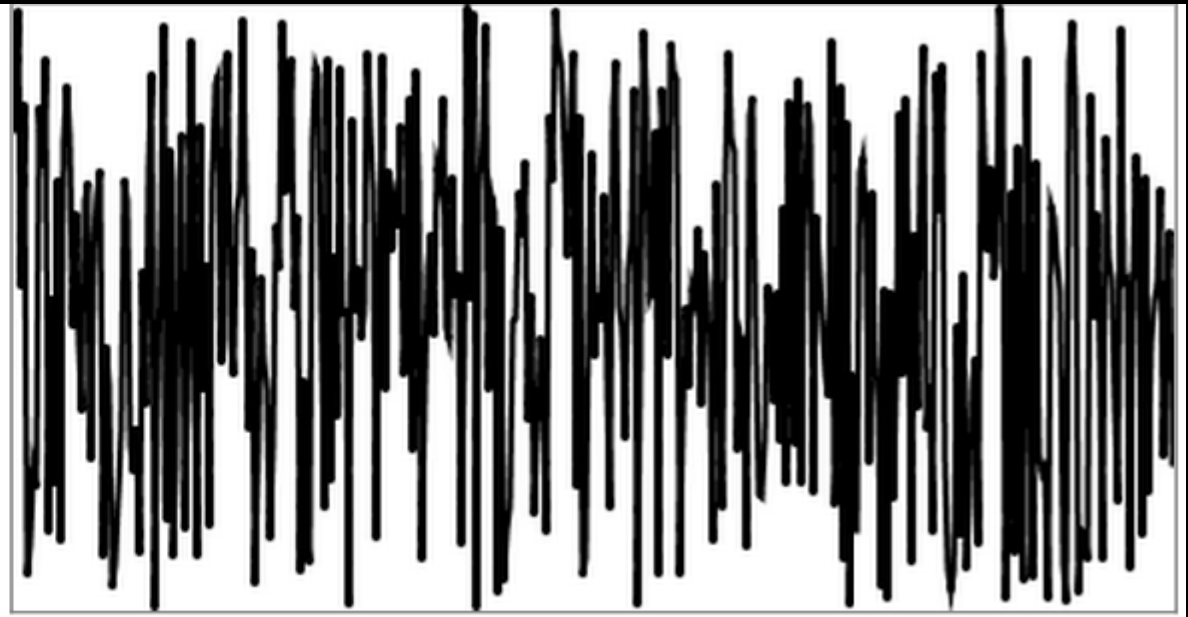

Figure I.6: Random

Perlin Noise Examples

Line in the Wind:
https://editor.p5js.org/scotchANDsolder/sketches/qYyiogwTF

Many Sticks in the Wind:
https://editor.p5js.org/scotchANDsolder/sketches/UIWEJOpX9

Perlin & Color :
https://editor.p5js.org/scotchANDsolder/sketches/N31w4TtxD
Spiky Germanium:
https://editor.p5js.org/scotchANDsolder/sketches/ts0QM2LQ_

Anatomy of Perlin Noise

noise(x) -> give you values between 0 & 1.0 (in a smooth way)

In order to use noise() we must ***scale*** & ***increment***.
For example, if we want perlin noise numbers between 0 and 200:

```
xpos = 200 * noise(off) //xpos will range between 0 & 200.0
off = off + 0.01 //off is updated
```

We are ***scaling*** by 200. This determines what range of numbers we want
We are ***incrementing*** by 0.01. This determines the smoothness. Lower the increment, more the smoothness.

Matter.js

Physics Libraries that simulates motions of objects in the real world.
Examples: https://brm.io/matter-js/

Matter.js

Steps to use:

1. Download template from here: http://palmerpaul.com/p5-matter/p5-matter-template.zip
2. Extract file and locate the following 2 files:
    1. p5-matter.min.js
    2. matter.min.js
3. Upload these to your p5.js web-editor. Click on the caret to see your files. Click on the arrow next to ‘Sketch File’ and upload the 2 files.
4. Edit index.html and add the following lines:
    - <script src="matter.min.js"></script>
    - <script src="p5-matter.min.js"></script>
5. Open and copy this script to your web editor: https://editor.p5js.org/scotchANDsolder/sketches/FAPjEKFcY

Matter.js

Available Shapes: Ball, Block, Barrier, Sign (Text)

1. Put matter.init() in the setup() function
2. Create shapes by calling matter.makeBall(.....) matter.makeBarreir(.....) etc.
   - Let's look at the parameters: http://palmerpaul.com/p5-matter/docs/#mattermakeball
   - Eg: let let ball = matter.makeBall(200,300,50)
   - myBall.show() -> displays and starts the physics engine for myBall
3. You can also add some features to each shape:
   - Eg: let ball = matter.makeBall(width / 2, 50, 60 , {restitution: 0.5 })
     - Restitution is the elasticity of the object
   - let myblock = matter.makeBarrier(width / 2,500,400,100, {restitution: 0.5, friction : 0.25 })
4. We can find out the location of each body (and other elements : http://palmerpaul.com/p5-matter/docs/#physicalobject)
   - Let xpos = ball.getPositionX()

Suggestions for Project 2

- Creative Coding aims to go beyond just representation/realism i.e. it lends itself to abstraction.
- Take an experimental approach i.e. try a few things before deciding which direction to take
- Have your imagery interact with the wall or space. Even simple textures that are mapped to a specific architecture/object can be interesting.
- Consider using random & perlin noise. Each visitor can have a unique experience
- Expand on Project 1 i.e. if you could animate your patterns how would they come alive?