Sound and Image Processing
Project Process Log
Project: Sound Synthesis

Idea
A simple sound synthesis and visual to represent the motion of falling water by using the sine wave oscillator provided in P5js.

Starter code
The starter code I use for this project is the one used in the third week of class:

```
let osc = null;

function setup() {
  createCanvas(400, 400);

  osc = new p5.SinOsc();
  osc.start();
}

function draw() {
  background(220);
  osc.freq(mouseX + 200);
}
```

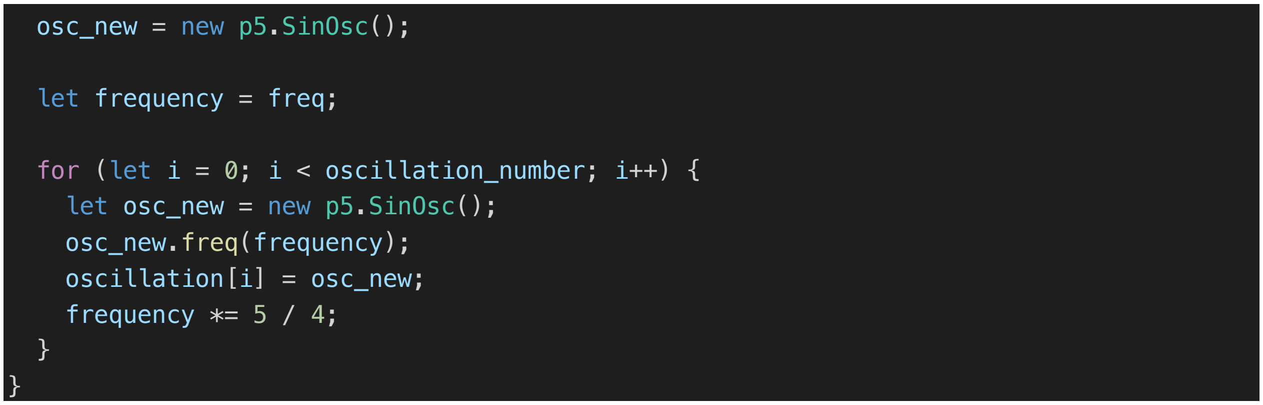
Then I add the nested loops for both sinewave and visual representation.
This process was a combination of trial and error and precalculated formula for the motion of ellipse going through the sinewave. Eventually, the motion of ellipses falling down achieved by this structure:

```
let theta = radians((oscillation_number - i + 3) * frameCount * 0.25);
let x = cos(theta) * tan(i + 6) * 10;
let y = tan(theta) * (i + 5) * 10;
```

Sound
The production of sound is also done through trial and error. Initially using the starter code as the base then manipulating the frequency and oscillation to achieve a sound that closely resembles water drips. (code below)

```
let oscillation = [];
let oscillation_number = 10;
let freq = 100;
let positions = Array(oscillation_number).fill(null);

function setup() {
  createCanvas(700, 800);
  noStroke();
```



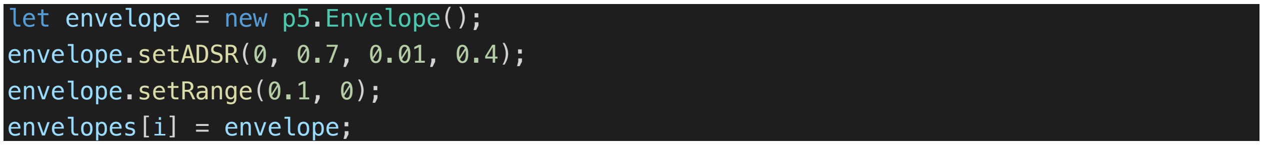
```
  osc_new = new p5.SinOsc();

  let frequency = freq;

  for (let i = 0; i < oscillation_number; i++) {
    let osc_new = new p5.SinOsc();
    osc_new.freq(frequency);
    oscillation[i] = osc_new;
    frequency *= 5 / 4;
  }
}
```

Conflict and Solution

The sound produced from the code above is not smooth and has a static “back sound”. Hence adding the set of code bellow to ensure that the sound produced is only within a specific range to avoid the static sound.



```
let envelope = new p5.Envelope();
envelope.setADSR(0, 0.7, 0.01, 0.4);
envelope.setRange(0.1, 0);
envelopes[i] = envelope;
```

This solution is provided from the third week of class.