Art of Sound

NMDE-404

Interactive IV

Lena Pang

Fall 2024

Project Overview

TIMELINE

5 weeks

TOPICS

Audio Visualization
JavaScript

PROMPT

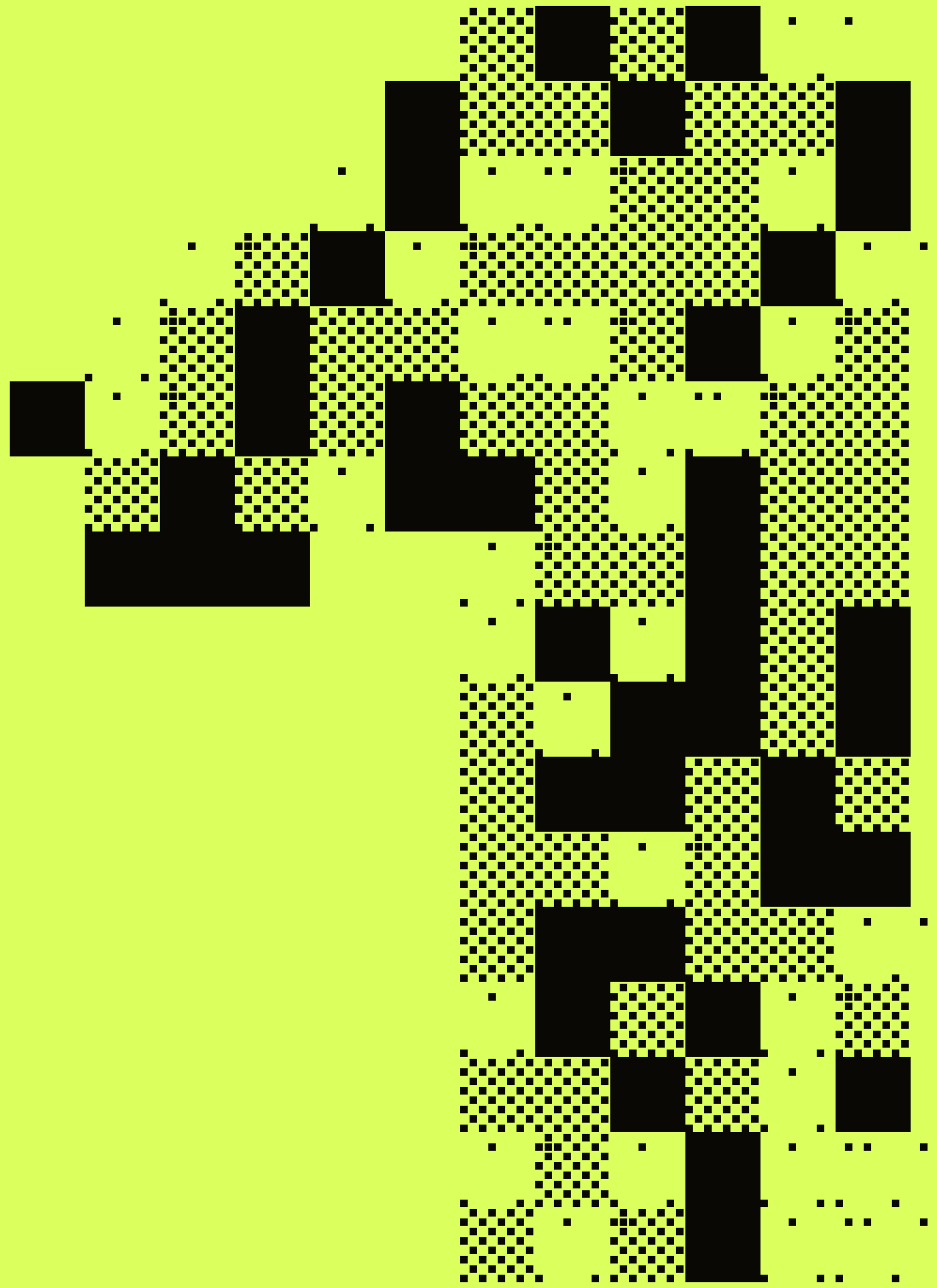
Choose a substantial topic and set deliverables for a passion project to showcase design and technical skills. This structure allows you to develop a portfolio piece that highlights your unique abilities and interests, tailored to your professional goals.

SUMMARY

Art of Sound is an audio–visualization experience in the browser, developed with JavaScript and the p5.js Sound library. It uses both wave amplitude and fast Fourier transform frequency distributions to process and display the sound data.

It features three modes, each of which offers an individual color scheme unique to that particular interpretation of the sound.

Table of Contents



Concept

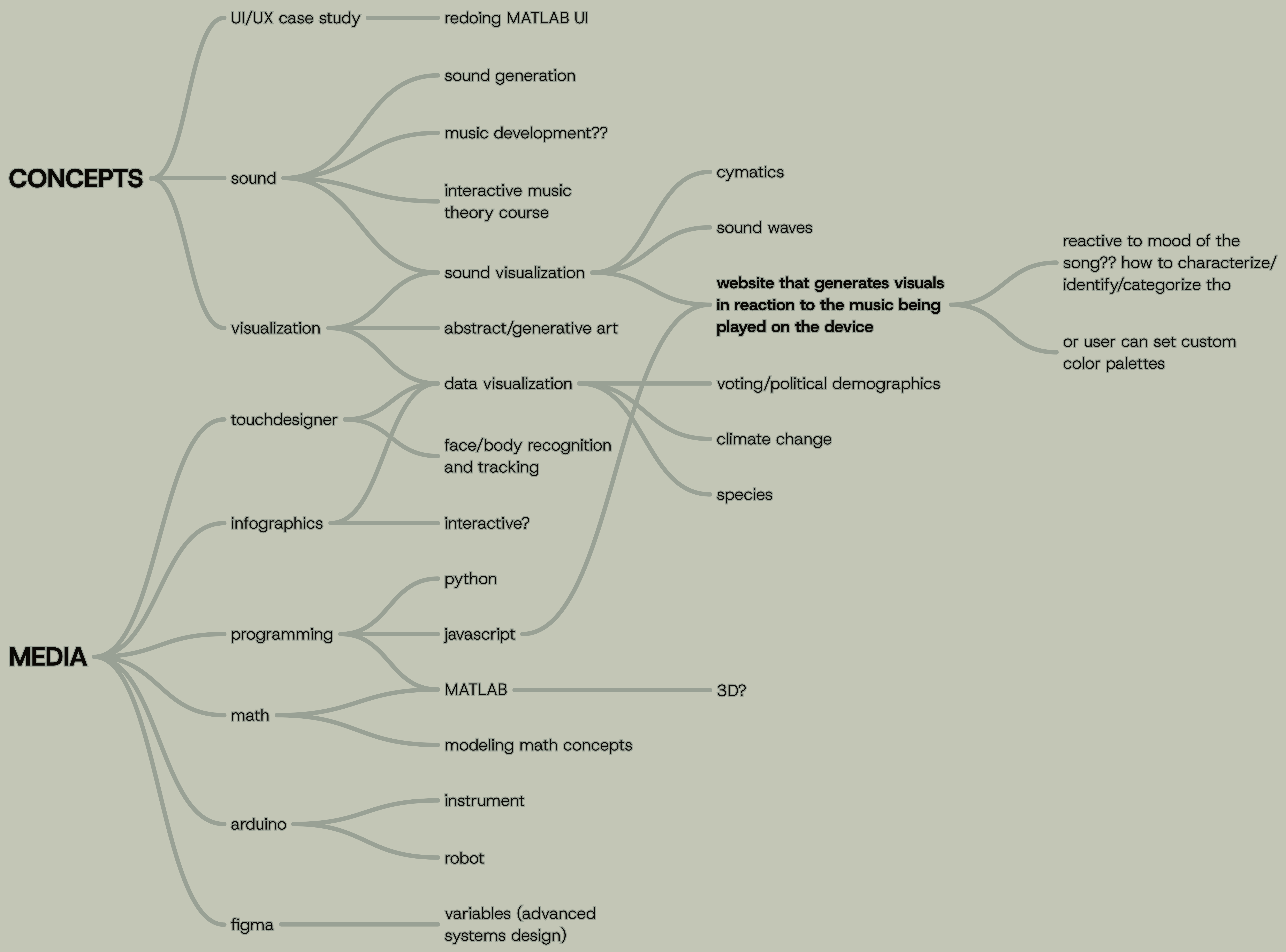
Brainstorming and establishing the topic, aesthetic, and scope of the project.

Brainstorming

During the ideation process, I created a mind map of a variety of concepts and media that I was interested in experimenting with.

From there, I found a few ideas that worked in the overlap of interesting concepts and interesting media, ultimately deciding on:

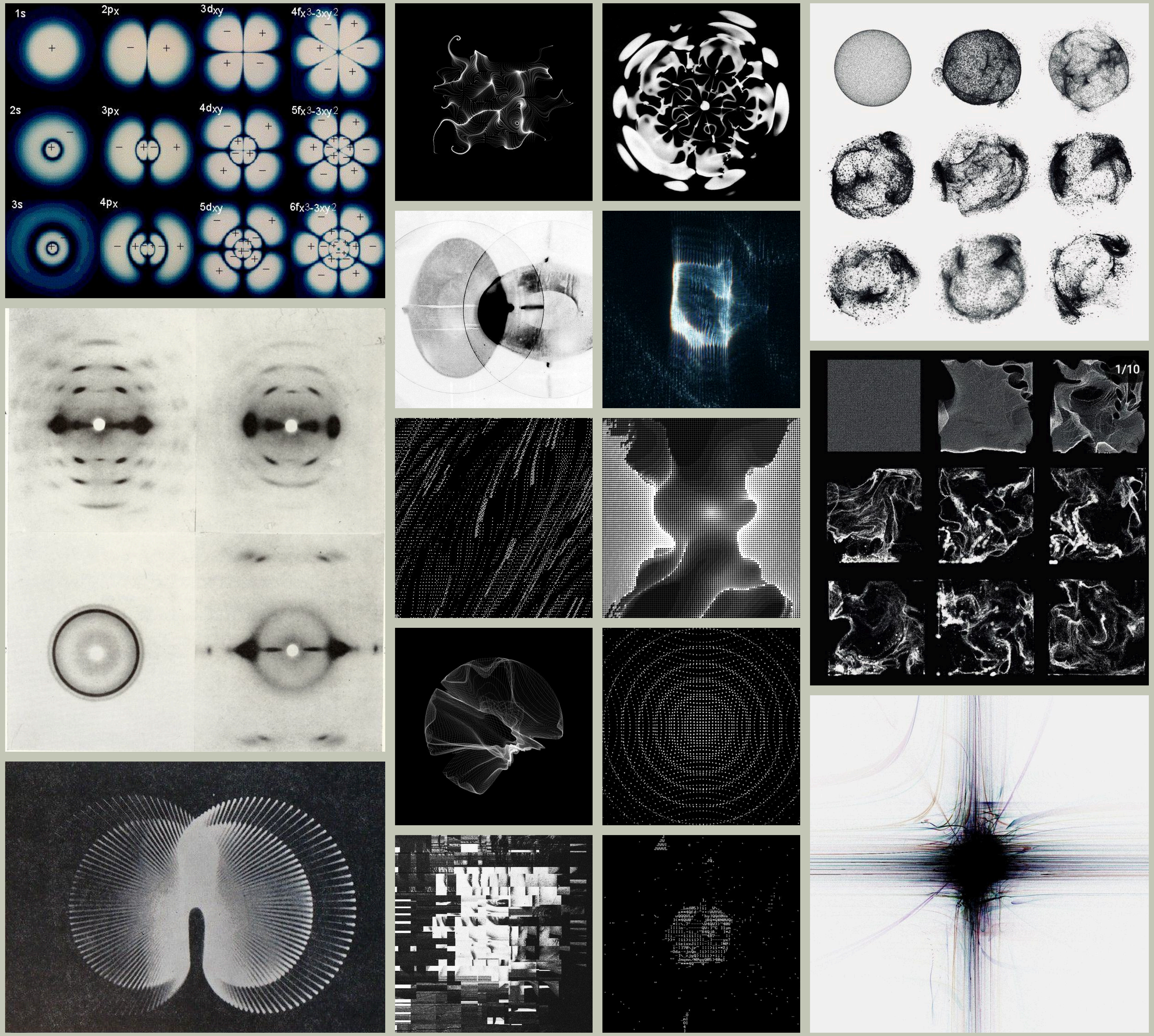
> **A website that generates visuals in reaction to the music or audio being played on the device.**



Moodboard

Taking aesthetic inspiration from scientific principles like magnetic fields, oscillating waves, velocity streamlines, particle distribution, and more to create a fascinating yet natural visualization of sound.

keywords: mathematics, patterns, graphics



Inspiration: Cymatics

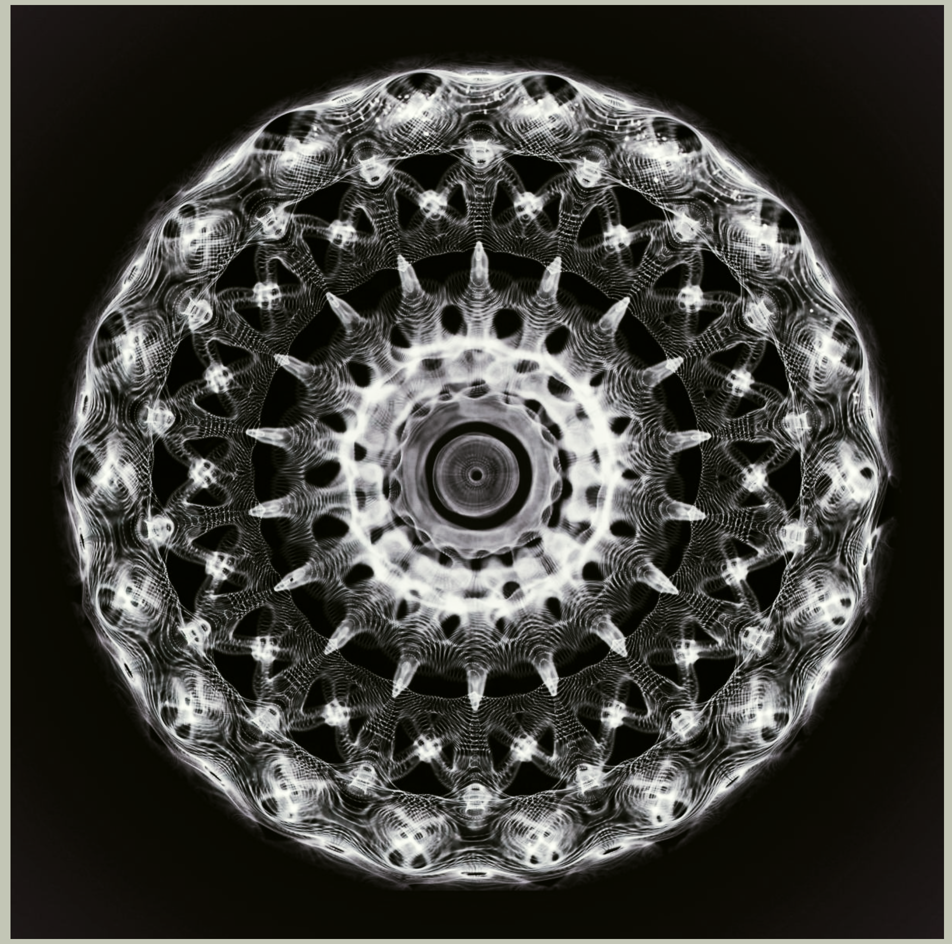
Cymatics is the study of visible sound.

When vibrated at certain frequencies, sand and water form geometric patterns.

Sand simply moves across the vibrating surface to form the patterns. Water, on the other hand, simply ripples—so it is easier to see the patterns by reflecting light on the water's surface.

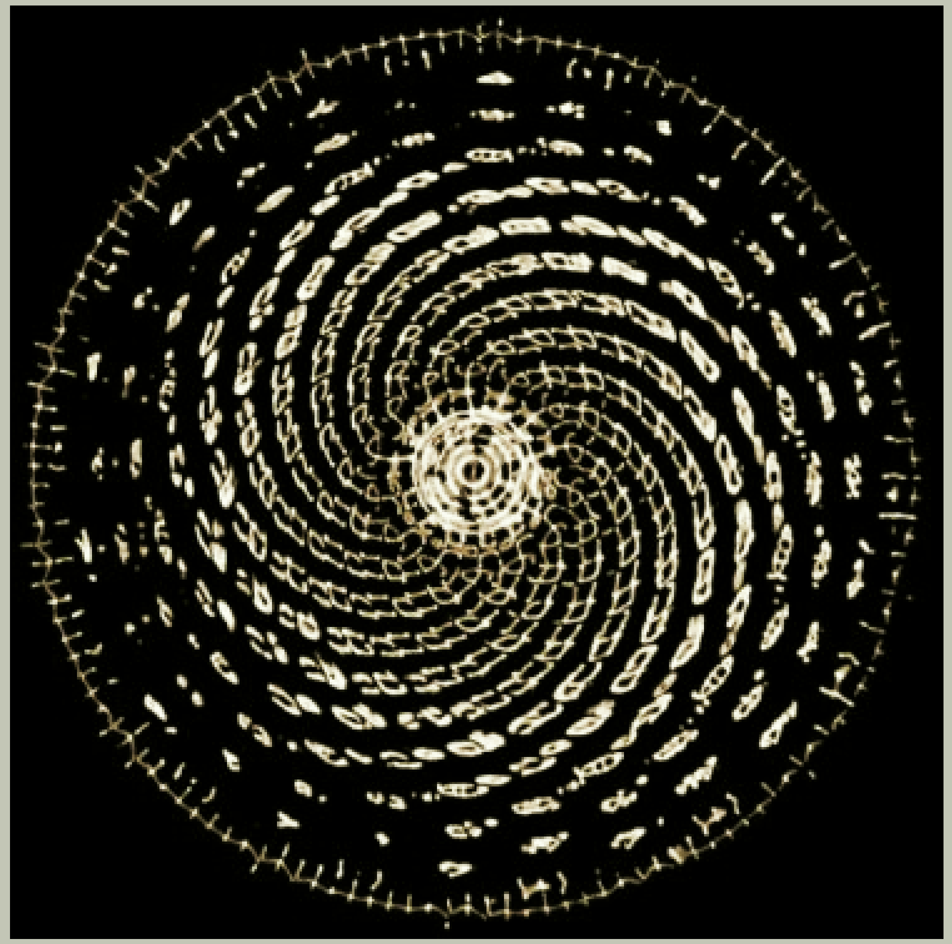
Many researchers have dedicated their lives to studying the underlying physics and mathematics of such a captivating phenomenon.

CymaScope.com

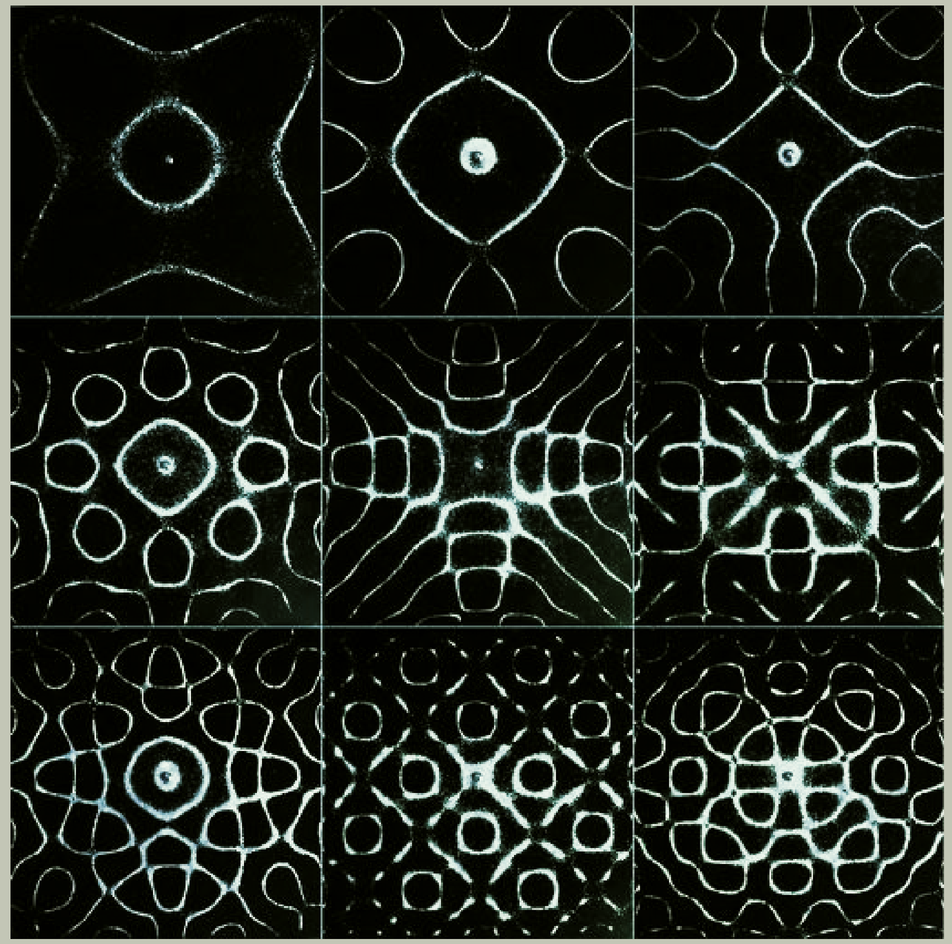


CymaScope.com

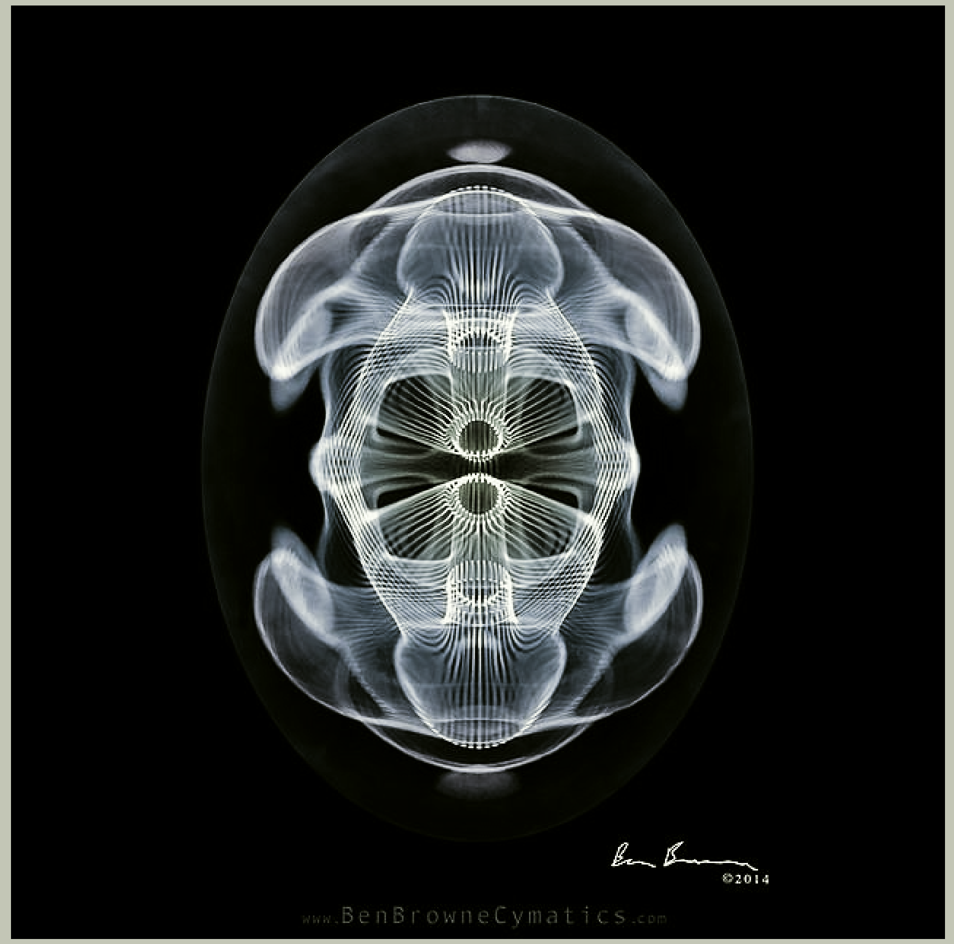
delamora.life/cymatics



delamora.life/cymatics



brusspup Resonance Experiment



BenBrowneCymatics.com

Development

Learning JavaScript, wireframing, testing program prototypes, and iterating through various designs.

Configuring Sound

Long story short, it is very difficult to get a program and/or website to be able to listen to the audio playing on a device.

I researched a variety of software and possible workarounds, ultimately coming back to my original plan to use JavaScript (specifically, p5.js). This allowed me to make a listening website.

But there's one major hang-up: **web pages can't listen to other tabs or applications; only the device microphone**.

Sound Input

Stereo Mix

- the audio "output" in Windows, can possibly be used as an input?

Virtual Sound Card

- recommendations
- Dante network
  - youtube.com/watch?v=vzzJQm-NGs8
- another one https://www.e2esoft.com/vsc/

JavaScript

p5.js Sound
https://www.youtube.com/watch?v=q2IDNkUws-A

- p5.AudioIn - security restriction in Chrome, must be httpS (or localhost)
  - Starting at Chrome 70 need to add: function touchStarted() {getAudioContext().resume(); }

  -
    ```
    var mic;
    function setup() {
          mic = new p5.AudioIn;
          mic.start();
          var col = mic.getLevel();
    ```
- static solution (if you want to run this only on a specific computer): use the listSource() function to try and identify your speakers ID to use in setSource(ID)

Tutorials

- youtube.com/watch?v=f0dwg99EVfo (mic based?)
- audio visualizer p5js (file based I think)
- shader programming
- p5js frequency analysis
- p5js sounds from embedded videos
- code

Other Software Possibilities

WinAmp

- big scene for visualizations
  - lots of add-ons
    - MilkDrop
    - there is also a platform independent version called projectM
- comes with the Advanced Visualization Studio; with these two you can build your own (AVS is much easier)

Morphyre Pro Music Visualization

- winamp add-on
- runs on its own
- will record sound from any running application
- ReadMe file on their website contains some information up the top on how to configure your Windows 7/Vista PC to record from Stereo Mix, which might be useful
- run WinAmp, select 'Open Location', type: linein:// (which actually records from the current record source)

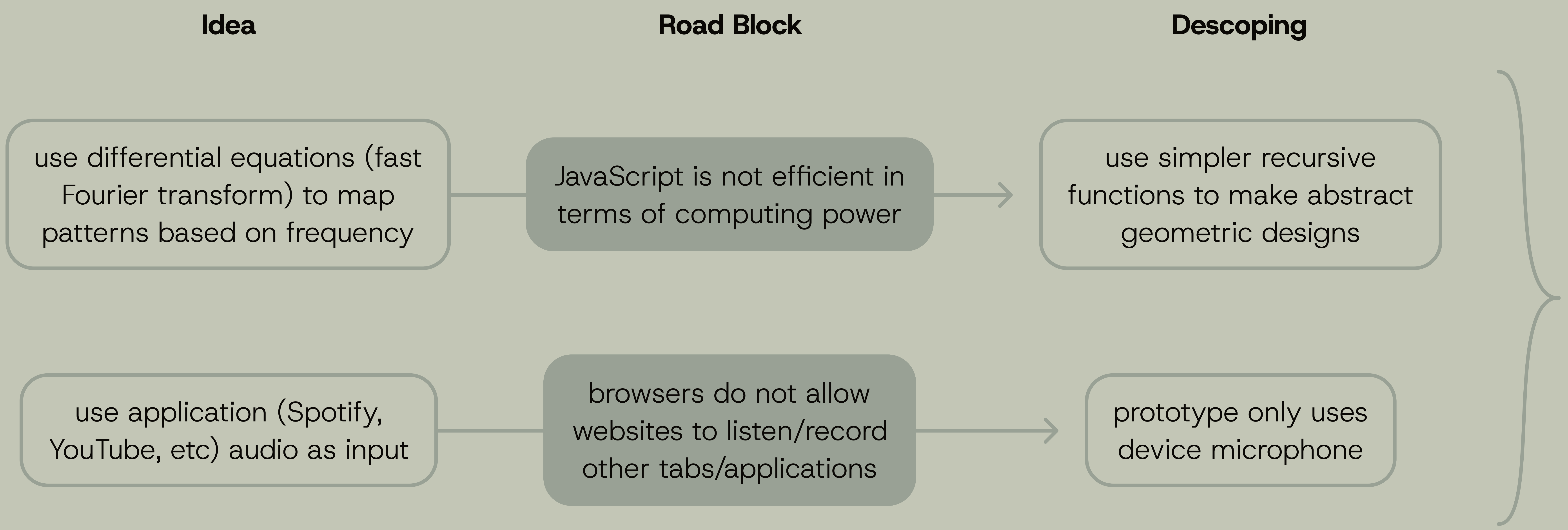
Pure Data

- TouchDesigner adjacent, sound visualizer
- https://www.youtube.com/watch?v=ULBP8iouw4I

Geissworks https://www.geisswerks.com/

- download stand-alone screensaver, which can react to the microphone, a CD, or your computer's stereo mix
- can drive any Winamp plug-in via a live (external) audio source - just hit CTRL+L (open location) and type in "linein://" (minus the quotes). Whatever is coming into your computer will show up in the visualizer.

Technical Limitations & Descoping



Idea

Road Block

Descoping

use differential equations (fast Fourier transform) to map patterns based on frequency

JavaScript is not efficient in terms of computing power

use simpler recursive functions to make abstract geometric designs

use application (Spotify, YouTube, etc) audio as input

browsers do not allow websites to listen/record other tabs/applications

prototype only uses device microphone

still satisfactory proof of concept!

emphasis is on the design & construction, rather than practical implementation

Processing Sound Data

Fortunately, the p5.js library has a **p5.Sound** add-on, which is capable efficiently of translating microphone sound data into a sequence of numbers in real time.

```
// Audio Setup

mic = new p5.AudioIn();

mic.connect();

mic.start();

getAudioContext().resume();
```

```
// Use Volume

let vol = mic.getLevel();
```

```
// Use Fast Fourier Transform

let fft = new p5.FFT(0.8,512);

let fourier = fft.analyze();
```

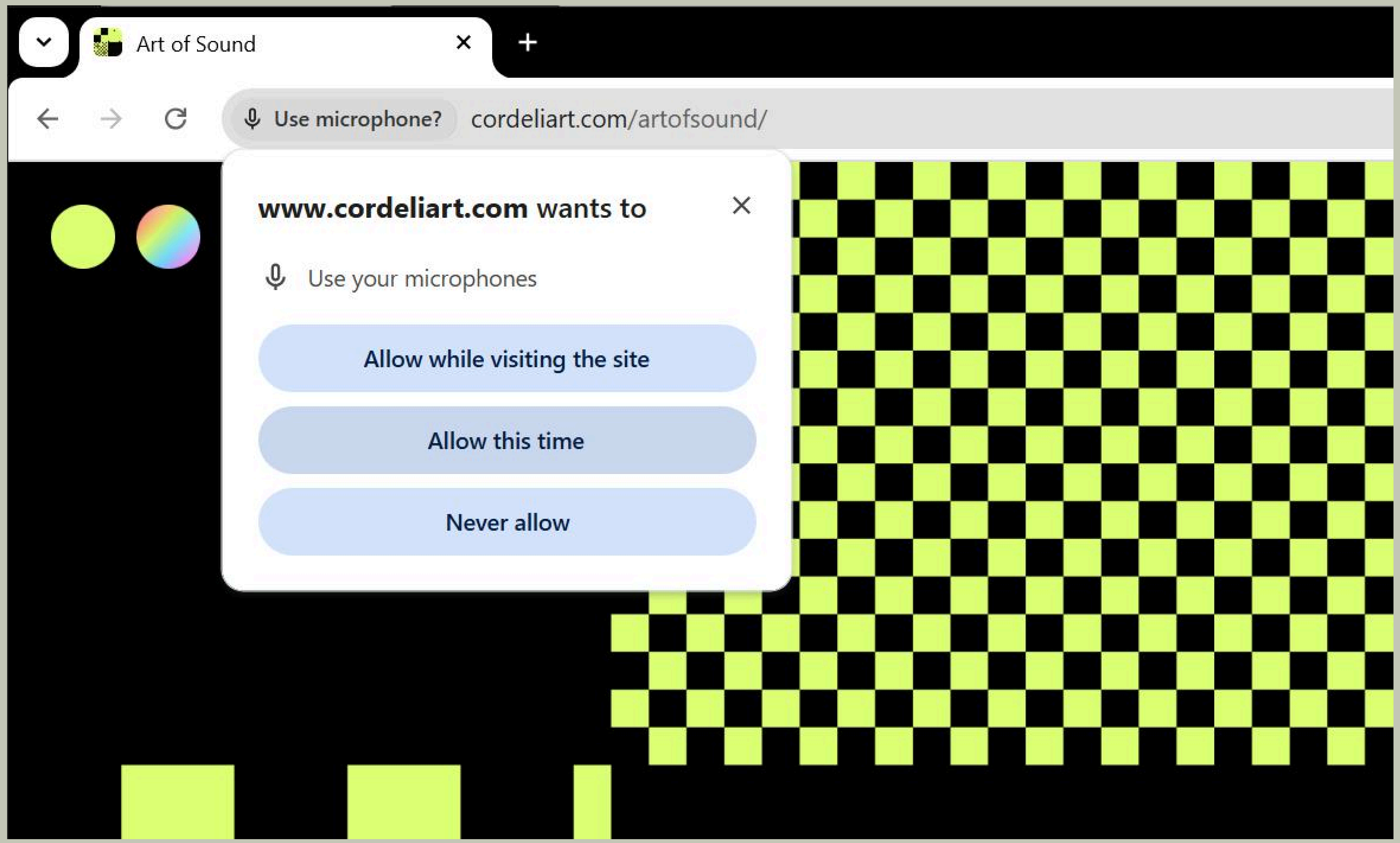
p5*js

Web Security

If it's not clear by now, browsers were not cooperative with my idea. There are several requirements to be able to use a microphone in Chrome 70 onwards:

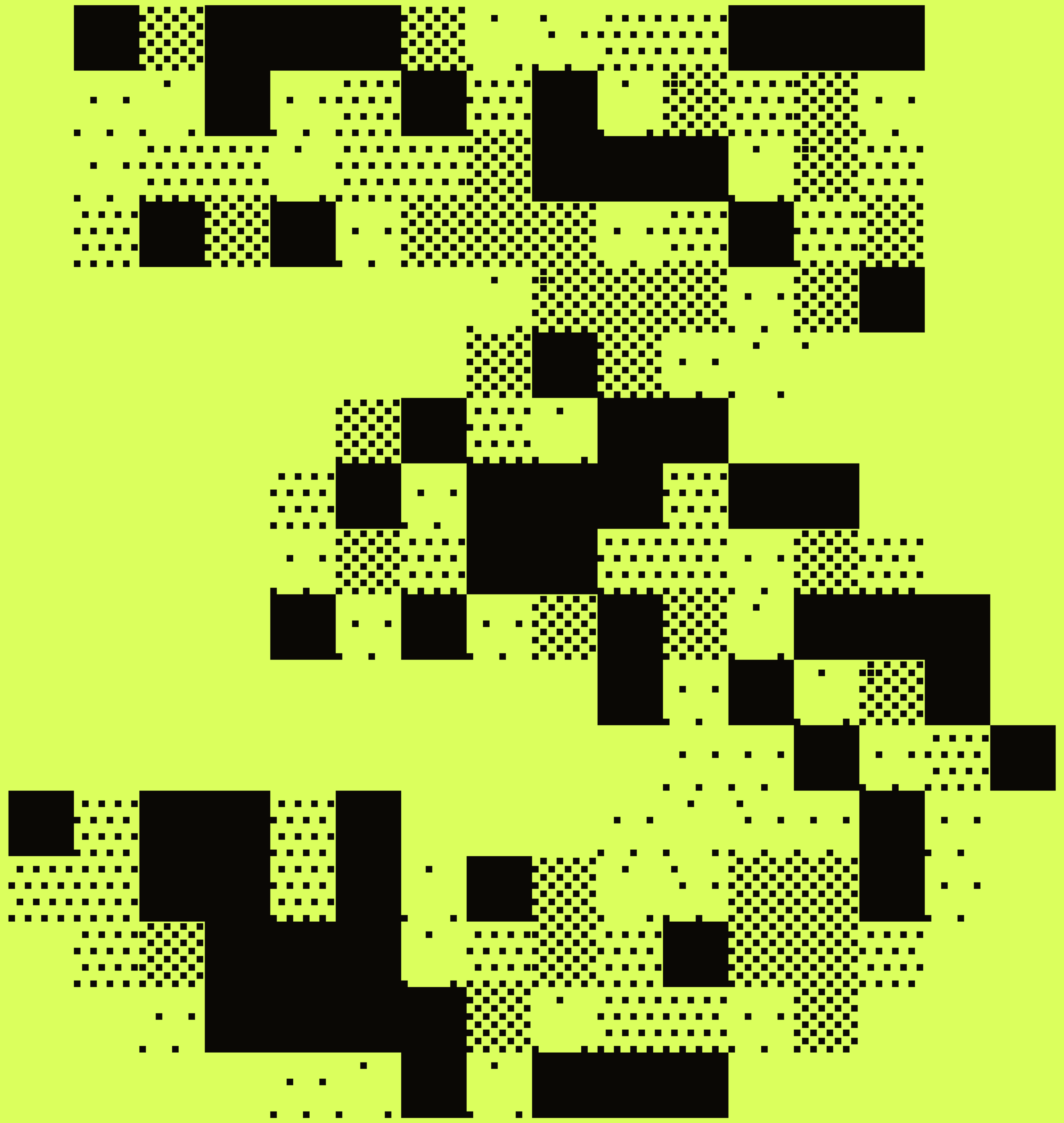
- the webpage must be https (not http)
  - localhost works but is not suitable for public use
- the user must interact with (touch/click) the page to prompt microphone permissions
- the user must allow microphone permissions

Even with all of that, activating the microphone still automatically silences any media players like YouTube and Spotify.



www.cordeliart.com wants to

Use your microphones



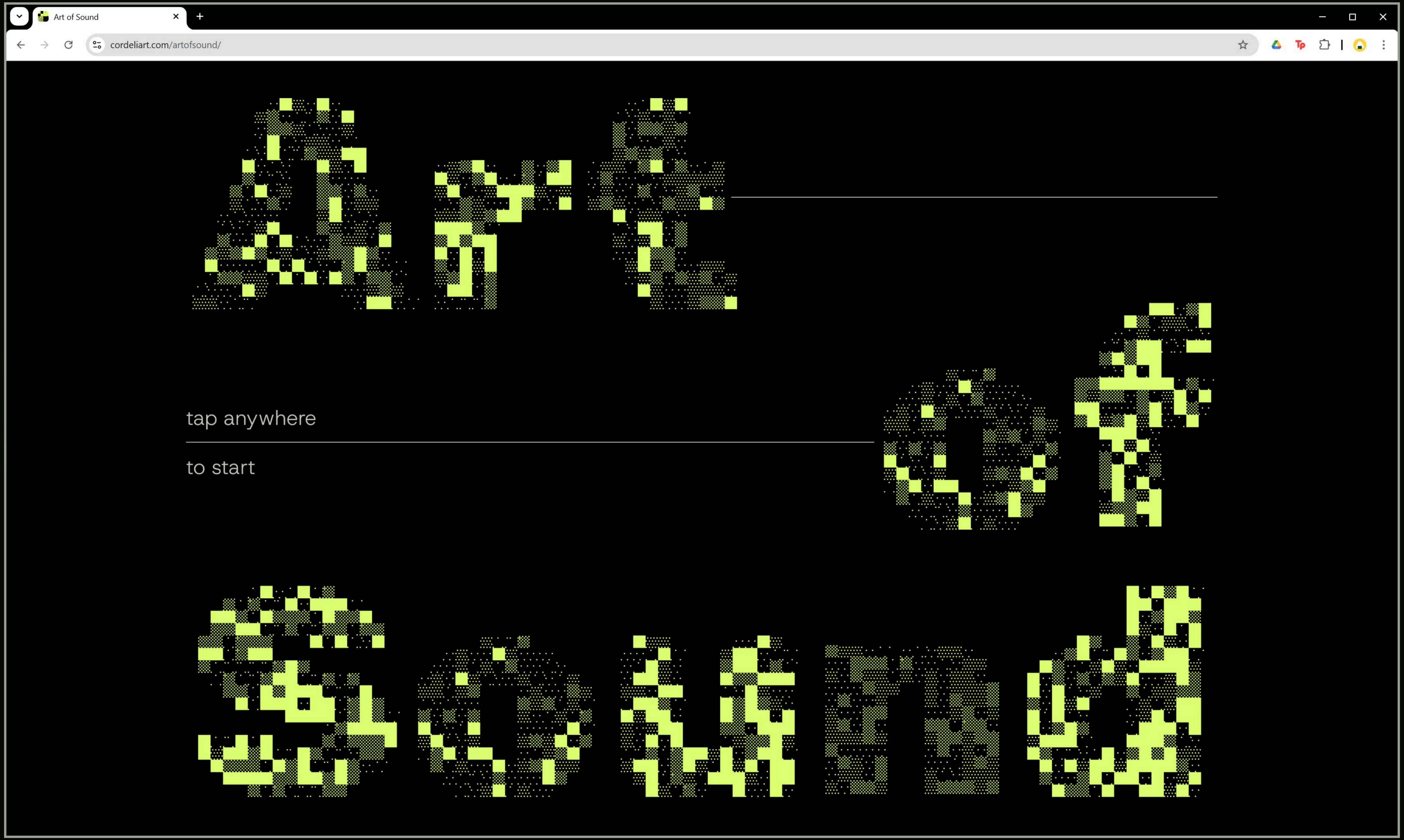
Final Product

Screencaps, layouts, style guide, algorithm overview, and designer reflection.

Test it out yourself!

cordeliart.com/artofsound

Launch Screen

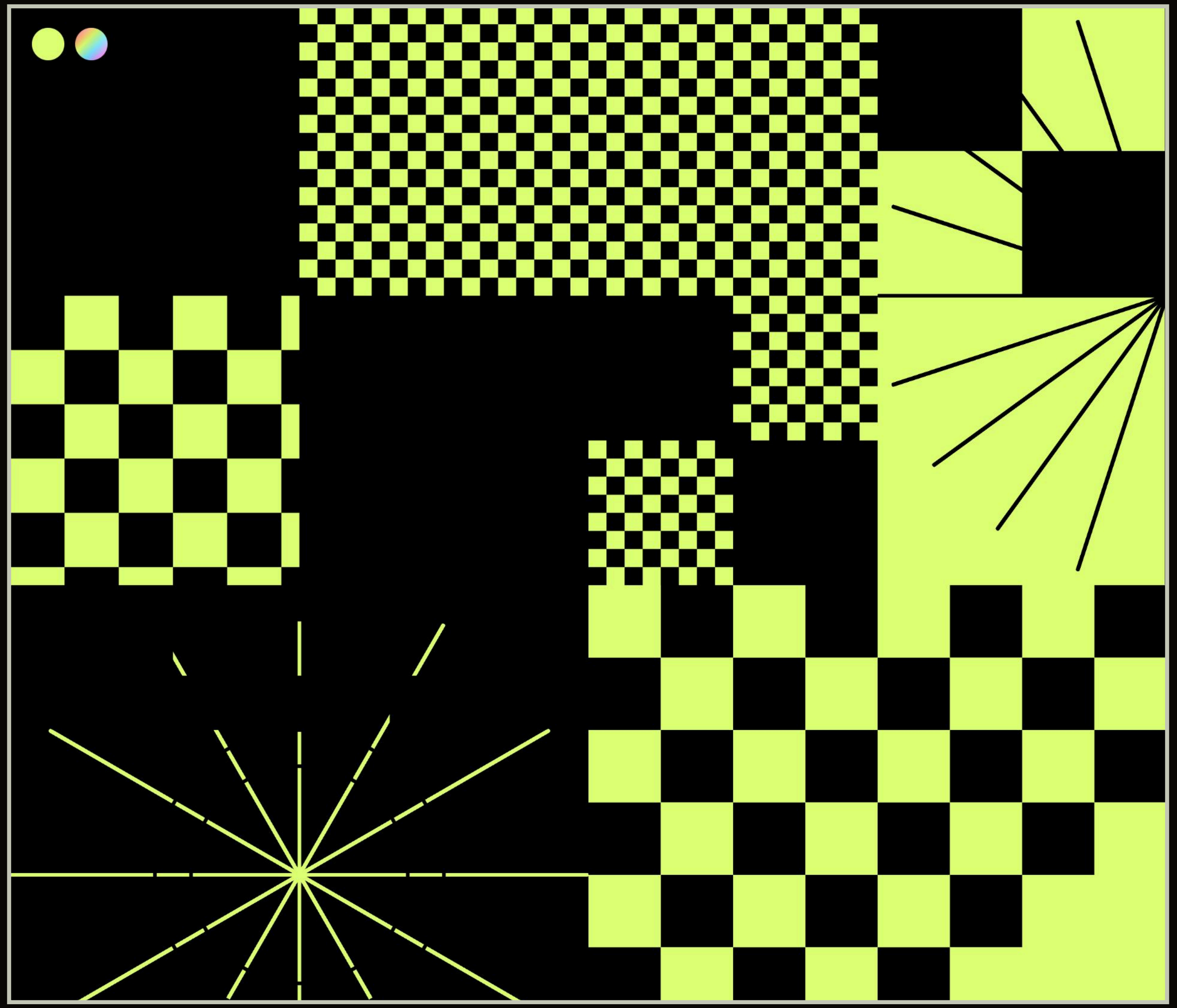


tap anywhere

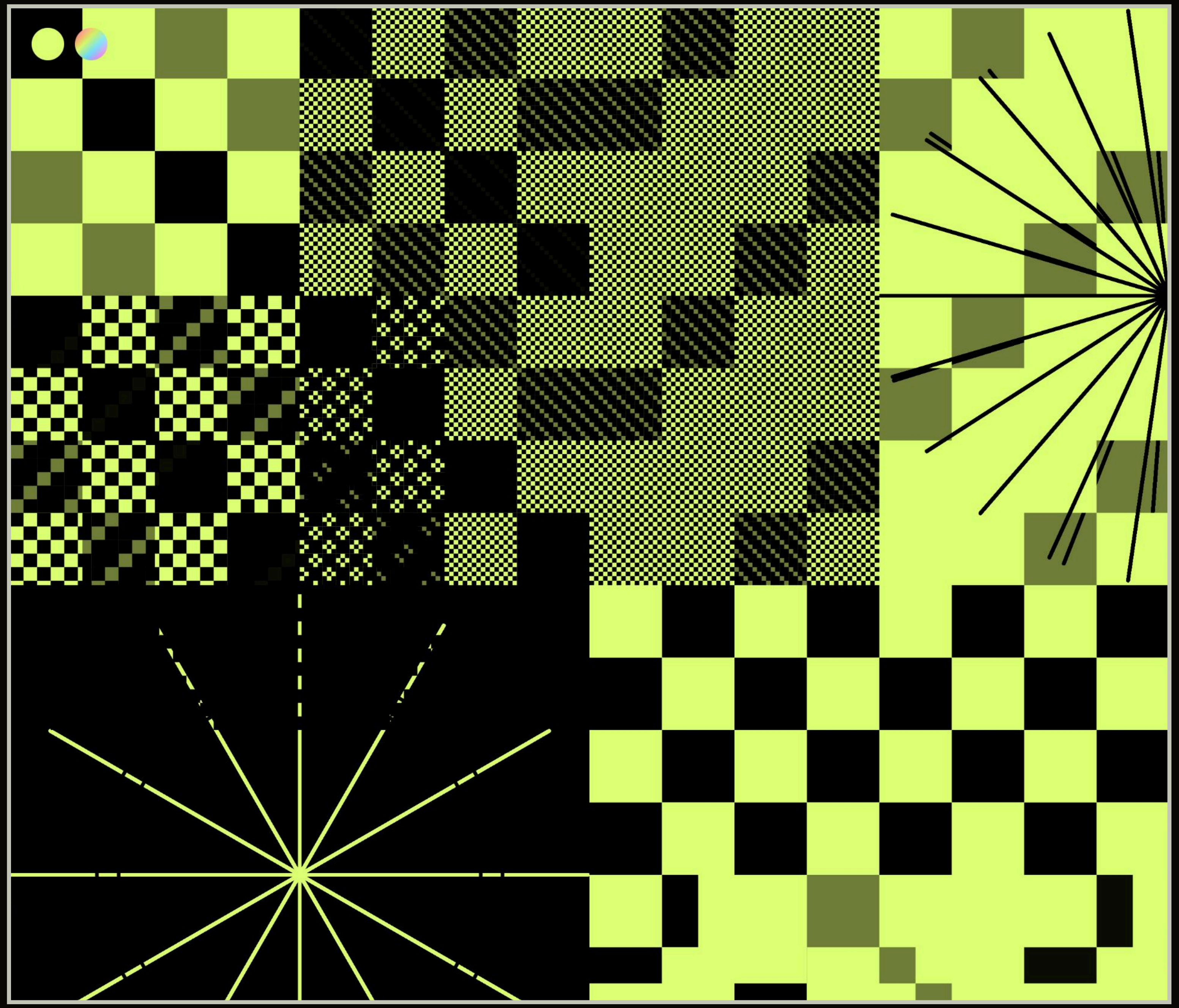
to start

Green

I'm a sucker for chartreuse—so the default mode after entry features a grid-based, geometric black-and-green design. The squares are reactive, based on the integer volume mod3 and mod4 with offsets, which allows for somewhat unpredictable and "randomly" independent movements between parts.



Idle

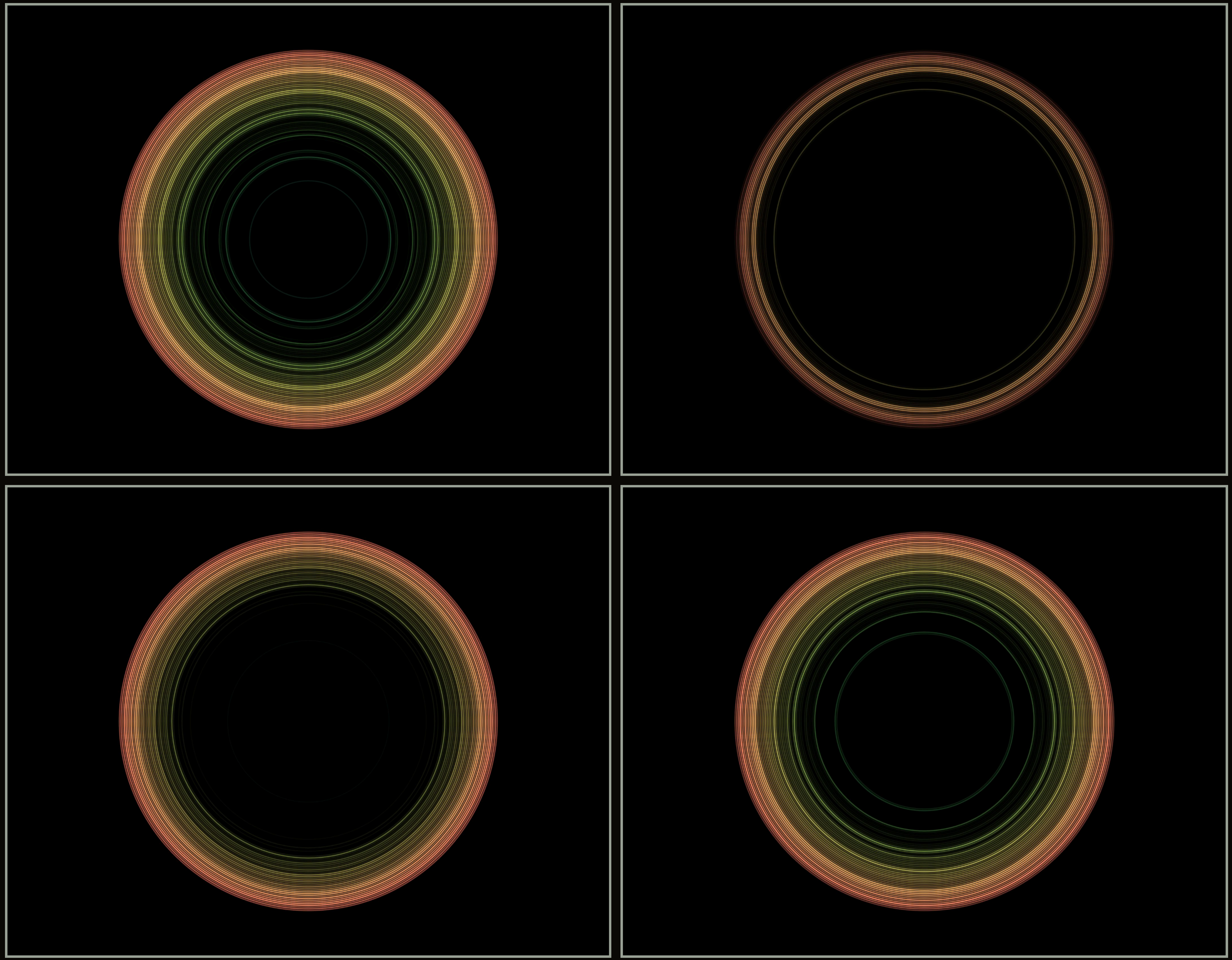


Active

Rainbow

This mode uses the Fast Fourier Transform function, which is an extremely fast algorithm that can deconstruct waveforms into pure sine and cosine functions.

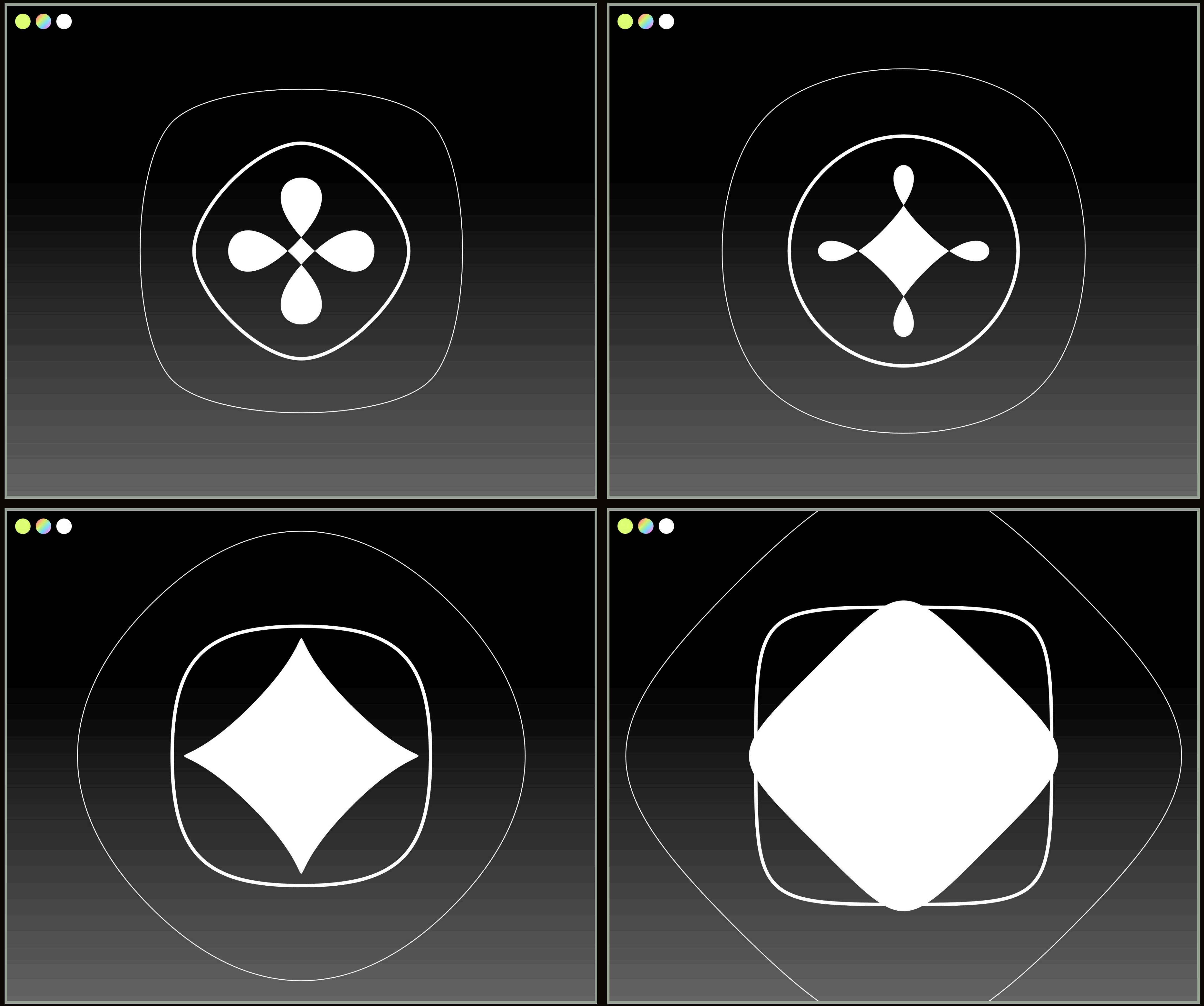
Low frequencies produce rings towards the outside, and high frequencies produce rings towards the inside.



White

This mode uses shapes made of bezier curves, where the positioning and pull of the vertices are reactive to volume.

Each shape uses slightly different strengths of the volume. As a result, the forms overlap and expand uniquely.



Style Guide

COLOR

#DCFF5E

#000000

#C3C7B6

#99A295

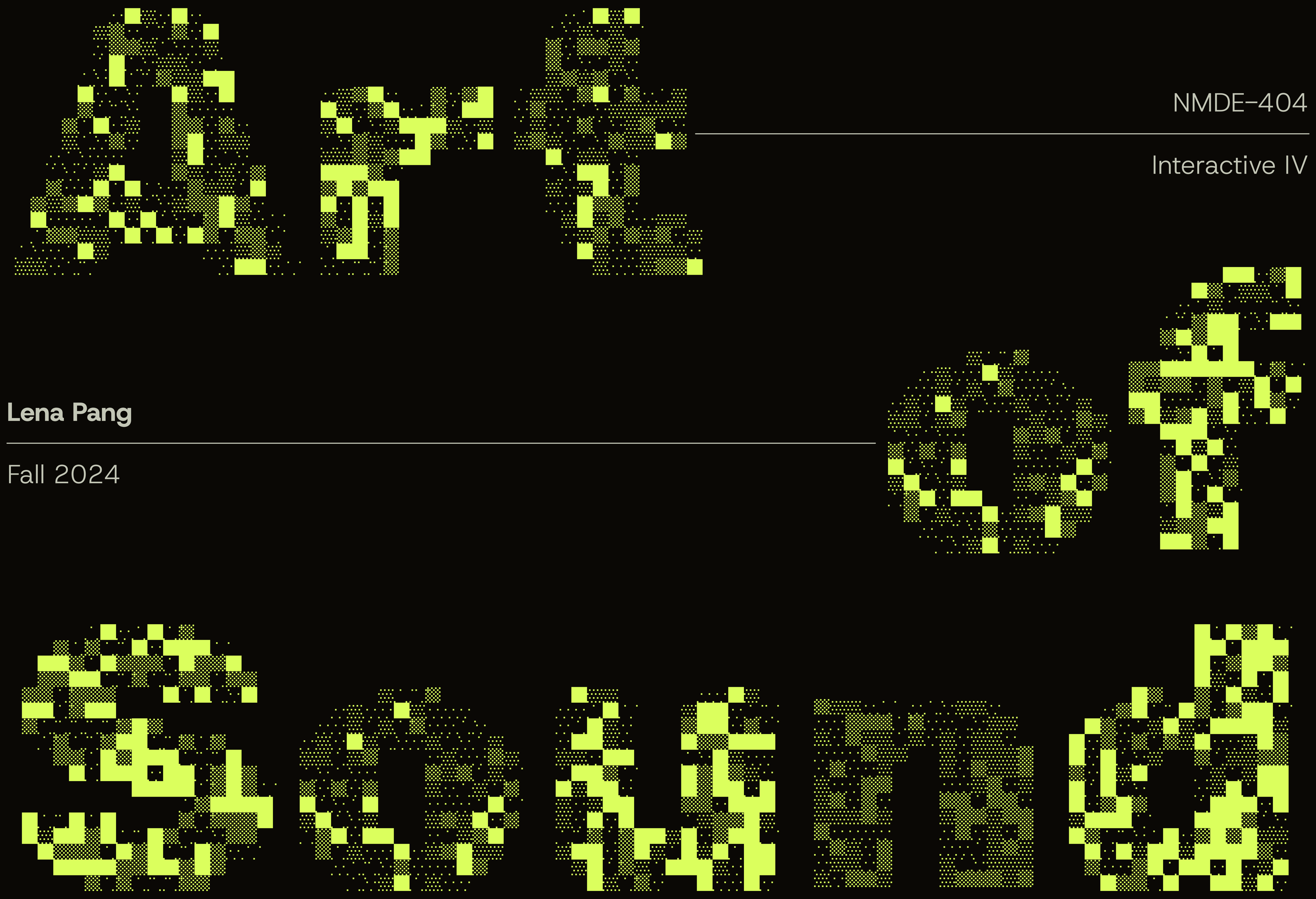
TYPE

Ag Rubik Pixels

Ag Host Grotesk Extra Bold

Ag Host Grotesk Light

Logo



NMDE–404

Interactive IV

Lena Pang

Fall 2024

Project Reel

https://youtu.be/jyBCaQWjkX0

Reflection

This project was filled with roadblocks: JavaScript real-time computational limits, browser security restrictions, copyright muting, you name it.

Still, I am happy with the final product. I wanted to experiment with visualizations of sound, and I got to do that through a variety of "modes" of the website, where each has a unique representation of sound. This also gave me the opportunity to work with JavaScript and libraries, which I enjoyed.

If I were to extend the project further, I would add more modes. I might also try looking into how to create an independently-run program that doesn't have the security and copyright overrides that browsers do. In the given time frame, that was simply too daunting of a task.

Resources

Learning:

p5js Sound

Patt Vira (YouTube)

The Coding Train (YouTube)

Reel Music:

Sad Piano Lost Love by WaveMaster (Pixabay)

https://pixabay.com/music/nostalgia-sad-piano-lost-love-30sec-198546/

Thanks for reading!

Hope you had a good time :)