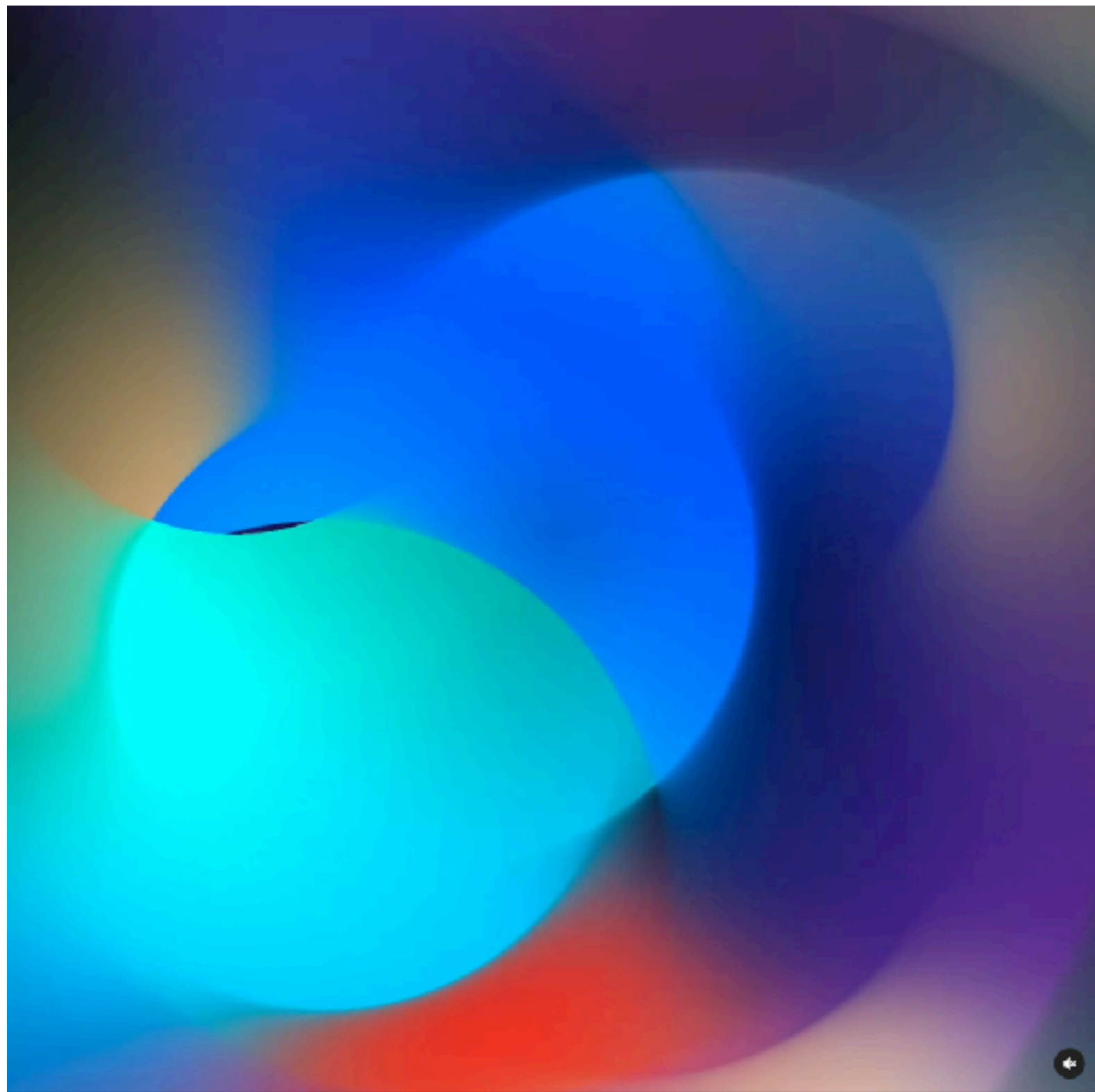


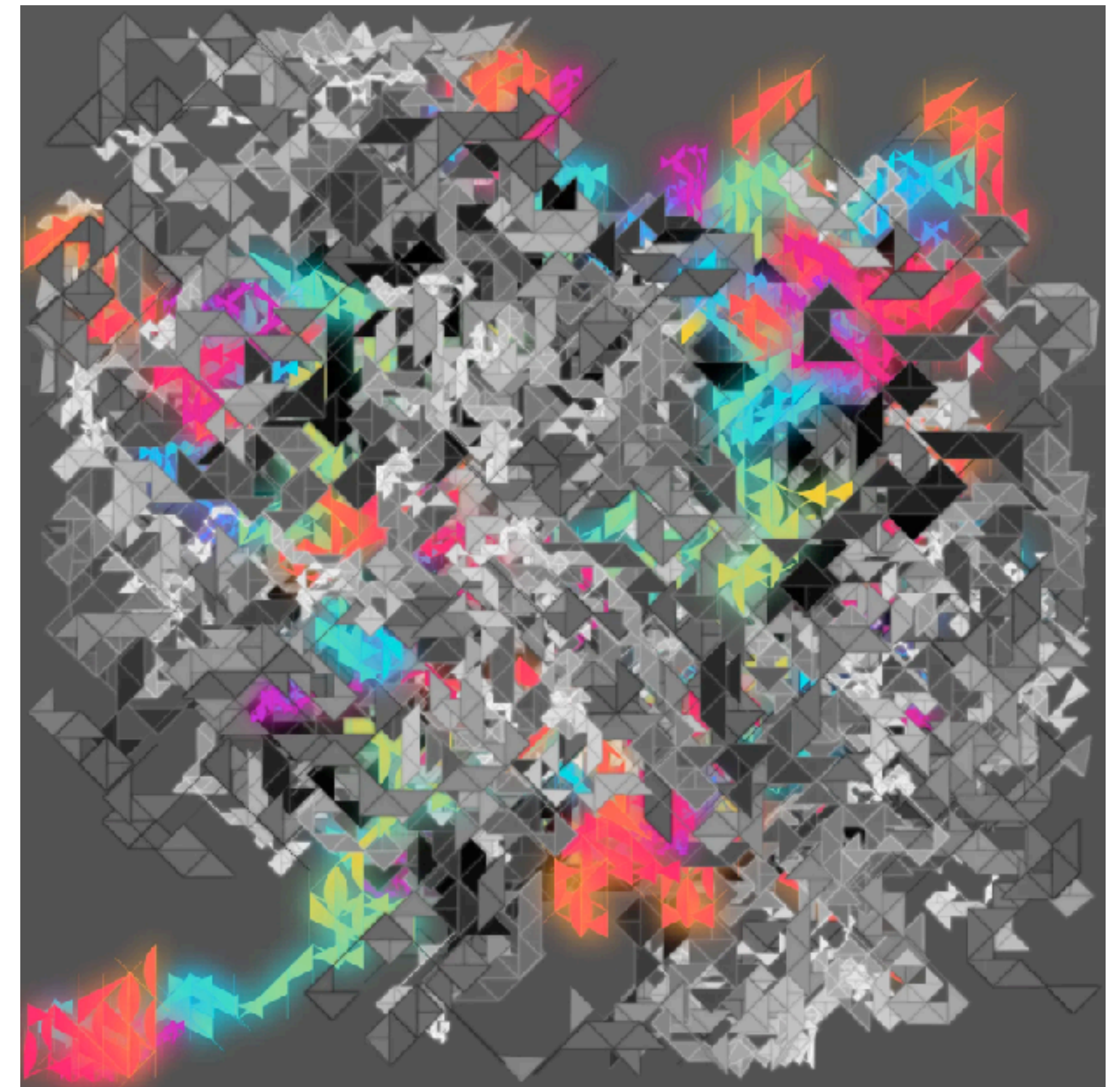
CS181DT Class 7: Creative Coding



Arcs by Zach Lieberman



Sketch Aquarium by teamLab



p5.js generative piece by shvemldr

Class 7 agenda

- Zipcrit
- Mini lecture: Creative coding
- Creative coding studio in p5.js

Announcements

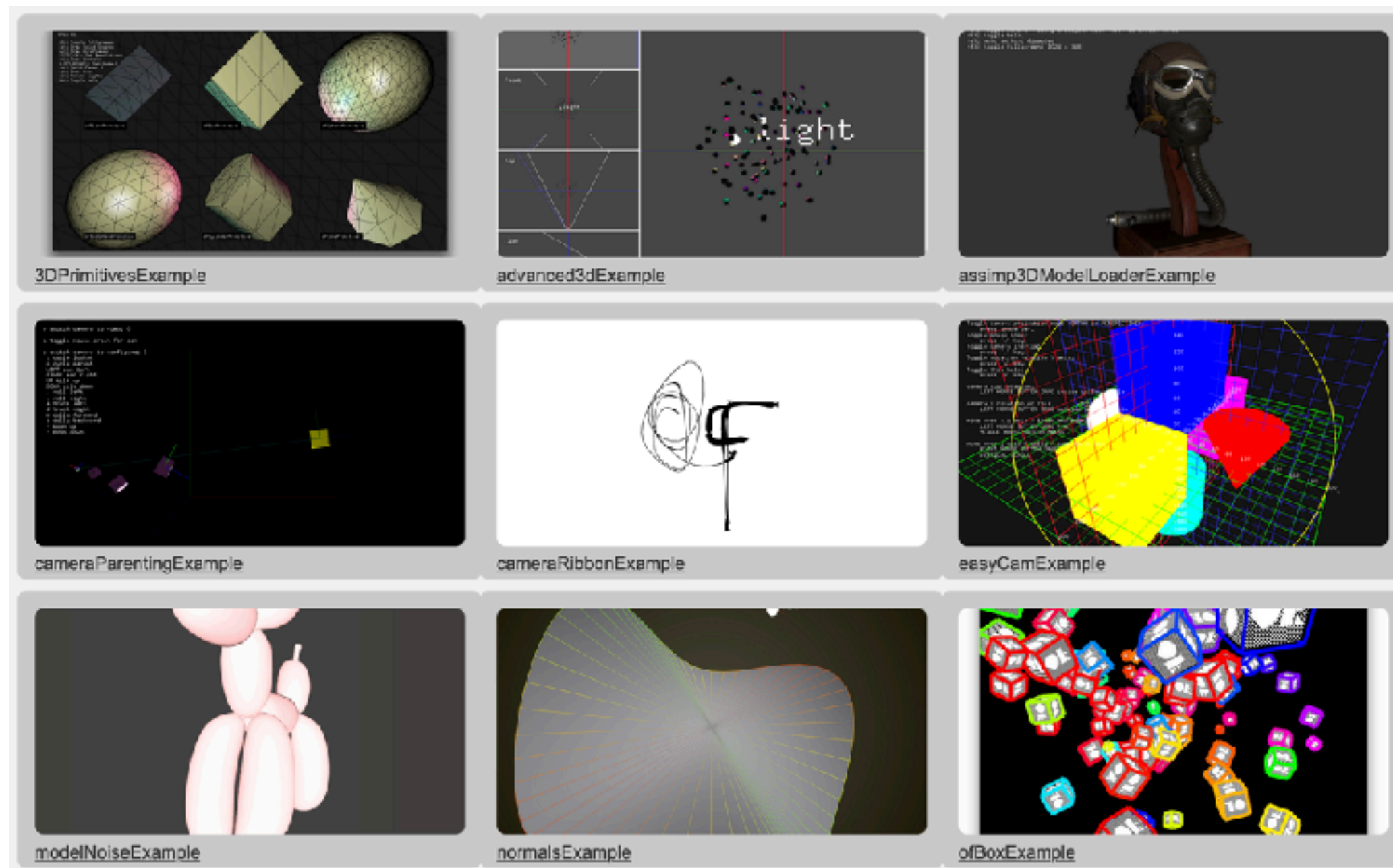
- How much computational material is in this class?
- You won't be learning new algorithms in class, but you will be coding a lot for your final project (and we'll go over strategies for structuring large software projects)
- We'll be programming in p5.js today and talking about how the *process* of creating art through programming differs from the *process* of creating art manually
- General theme: humanistic perspective on technical material

Intro to Creative Coding

Creative coding

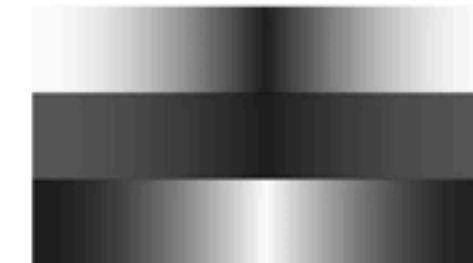
- Code written to be *expressive* rather than functional
- Many “domain specific languages” (DSLs), such as...

Compared to non-coding digital art tools (like Photoshop), what kinds of art can only be made with code?

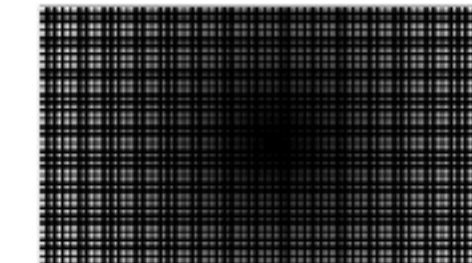


openFrameworks (C++)

Arrays



Array

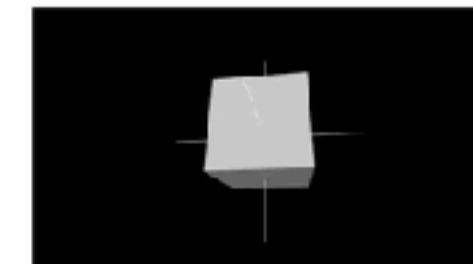


Array 2D



Array Objects

Camera



Move Eye



Orthographic



Perspective

Color



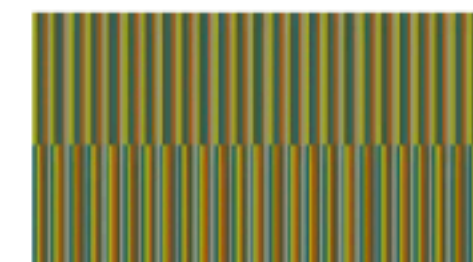
Brightness



Color Variables



Hue



Relativity



Saturation

Processing.py

p5.js

Processing
for Android

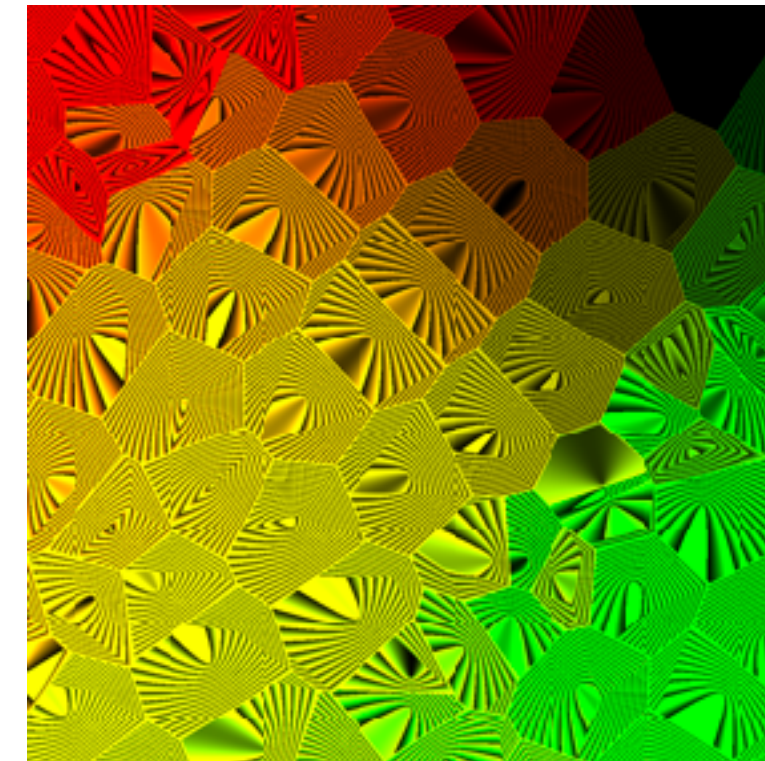
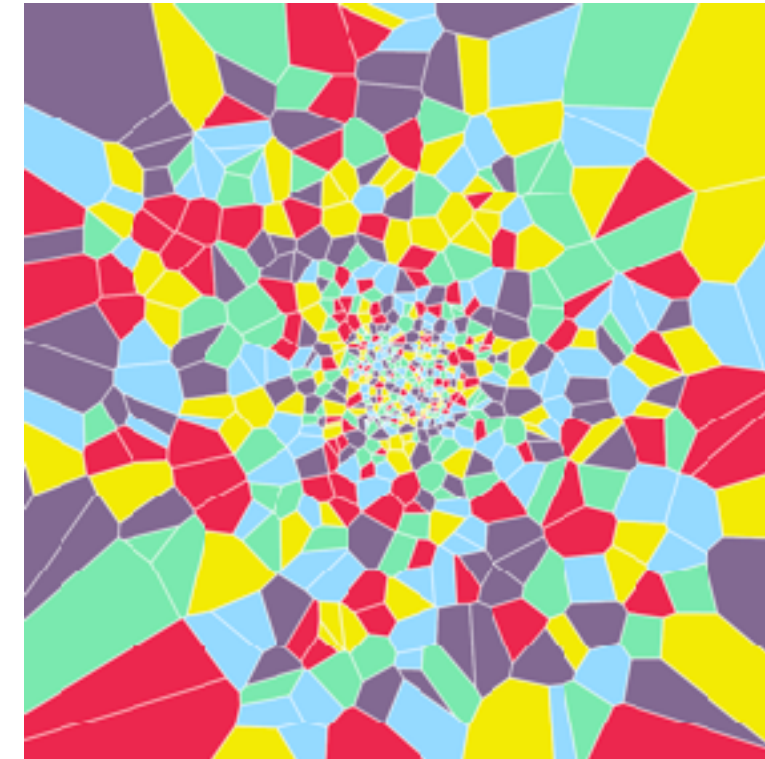
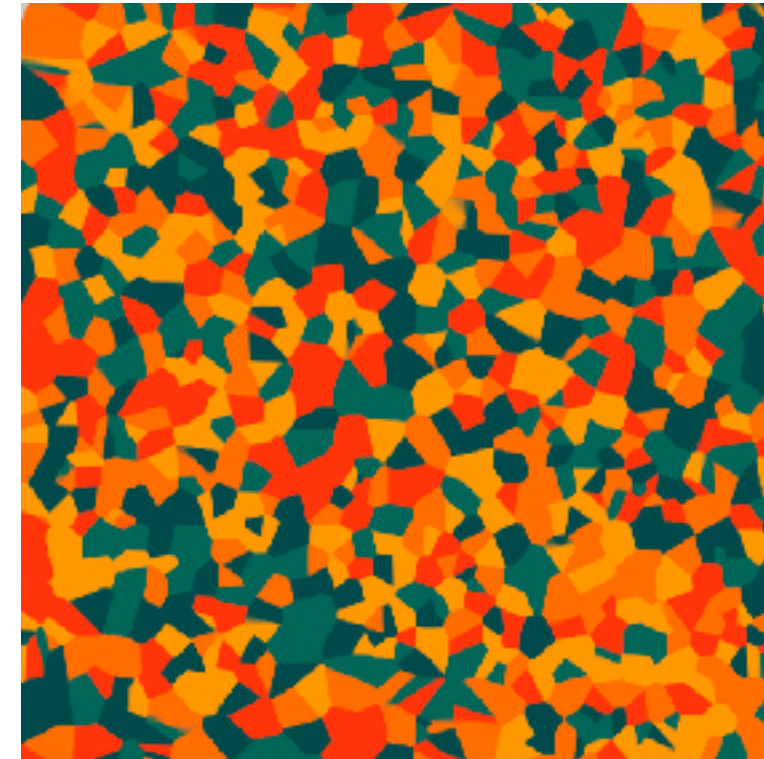
Processing
for Pi

Processing (Java)

Use cases



Interactive Art



Algorithmic art



Video Jockeying (VJing)

Community oriented

- “From the beginning, Processing was designed to be as simple as possible for beginners, knowing that its simplicity would also benefit more experienced users as well.”
- “[...] to empower people of all interests and backgrounds to learn how to program and make creative work with code, especially those who might not otherwise have access to these tools and resources.”

The Processing Foundation's core principles of accessibility and empowerment are facilitated through our Advocacy Program. We seek out and work with groups who have historically not had access to the fields of technology, code, and art, whether because of race, gender, class, sexuality, and/or disability. By facilitating dialogue and collaboration, the Foundation sponsors and hosts events that aim to broaden the audience for our software projects, welcome those new to the community, and nurture the specific needs of different groups.

Processing Community Survey 2022

We want to hear from you! To improve our tools and community experience, we need your feedback. If you use Processing or p5.js, or any of the tools from the Processing family, this is your chance to tell us about your experience and help guide future

Please respond by Dec 11th

Processing | Download | Documentation | Learn | About | Search

Tutorials

Video Tutorials

Links to videos that cover the Processing basics.

Hello Processing
by Daniel Shiffman et al.
A quick intro guiding you to create a simple drawing tool. No software install needed.

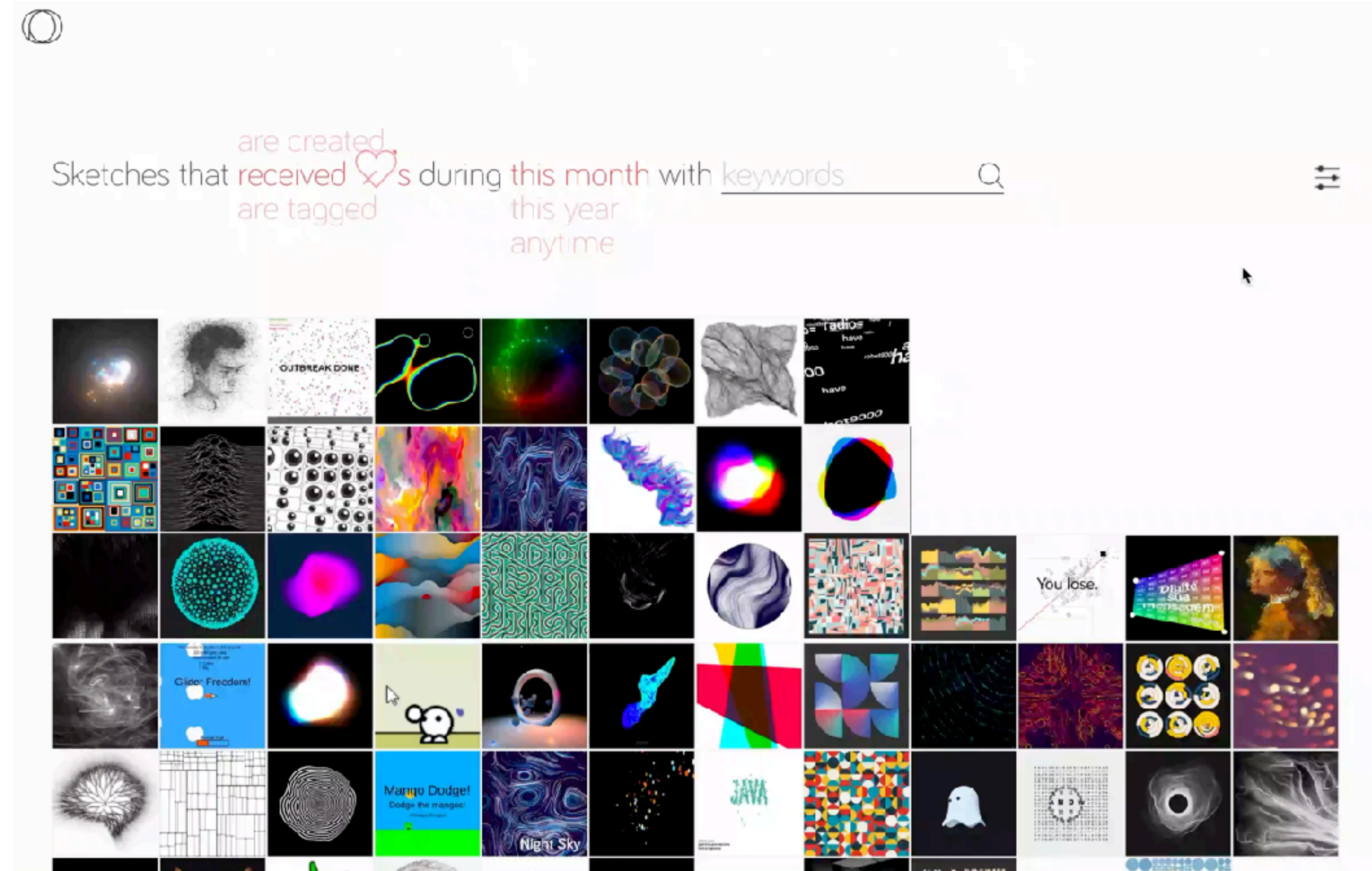
Learning Processing
by Daniel Shiffman
A complete introductory course on Processing, designed for complete beginners.

Debug
by Daniel Shiffman
Learn how to pause and step through your running code with the Processing Debugger.

The Coding Train
by Daniel Shiffman
All aboard the Coding Train with Daniel Shiffman, a YouTube channel dedicated to beginner-friendly creative coding tutorials and challenges.

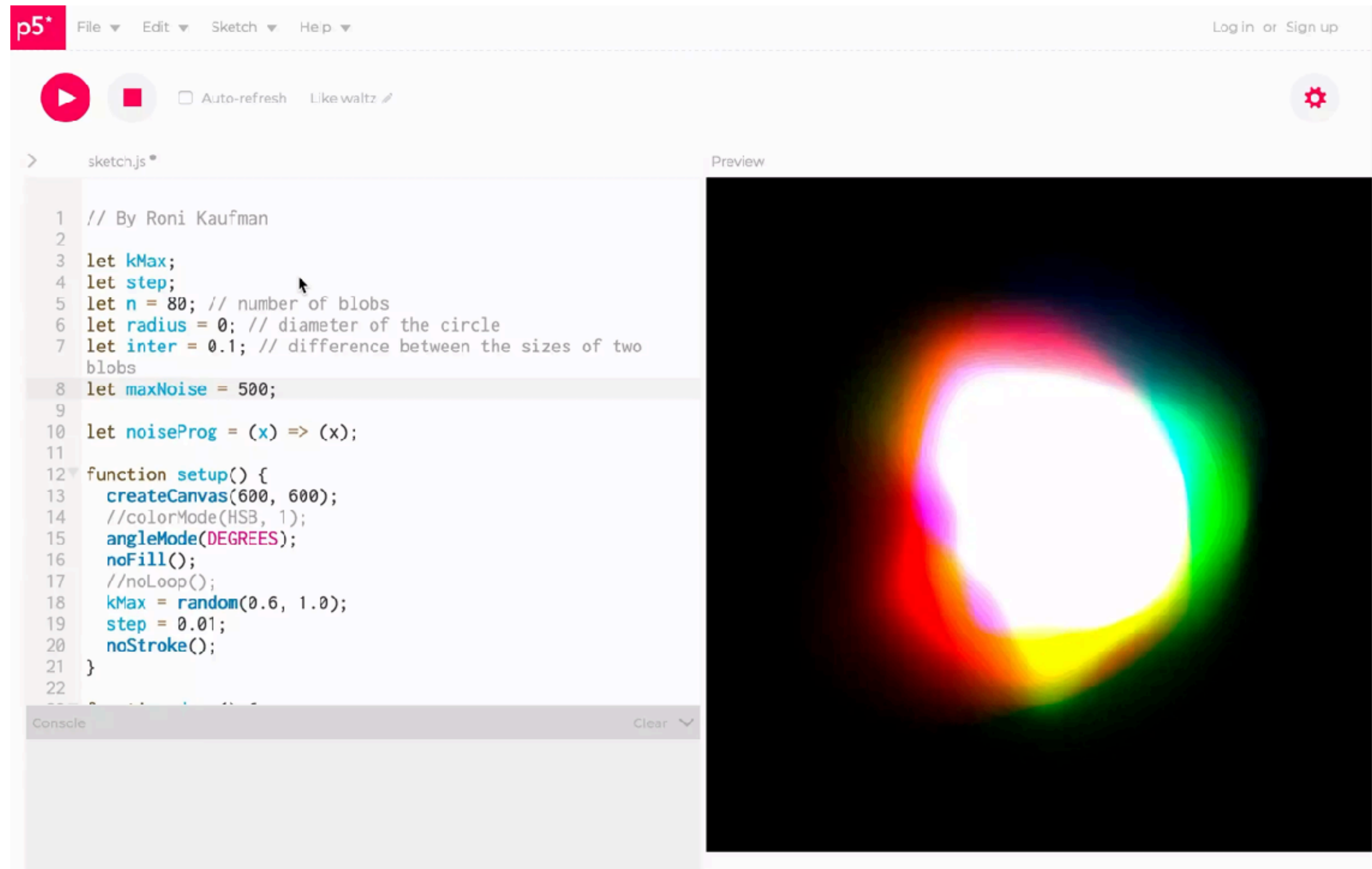
Accessible to new programmers

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



Accessible to new programmers

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



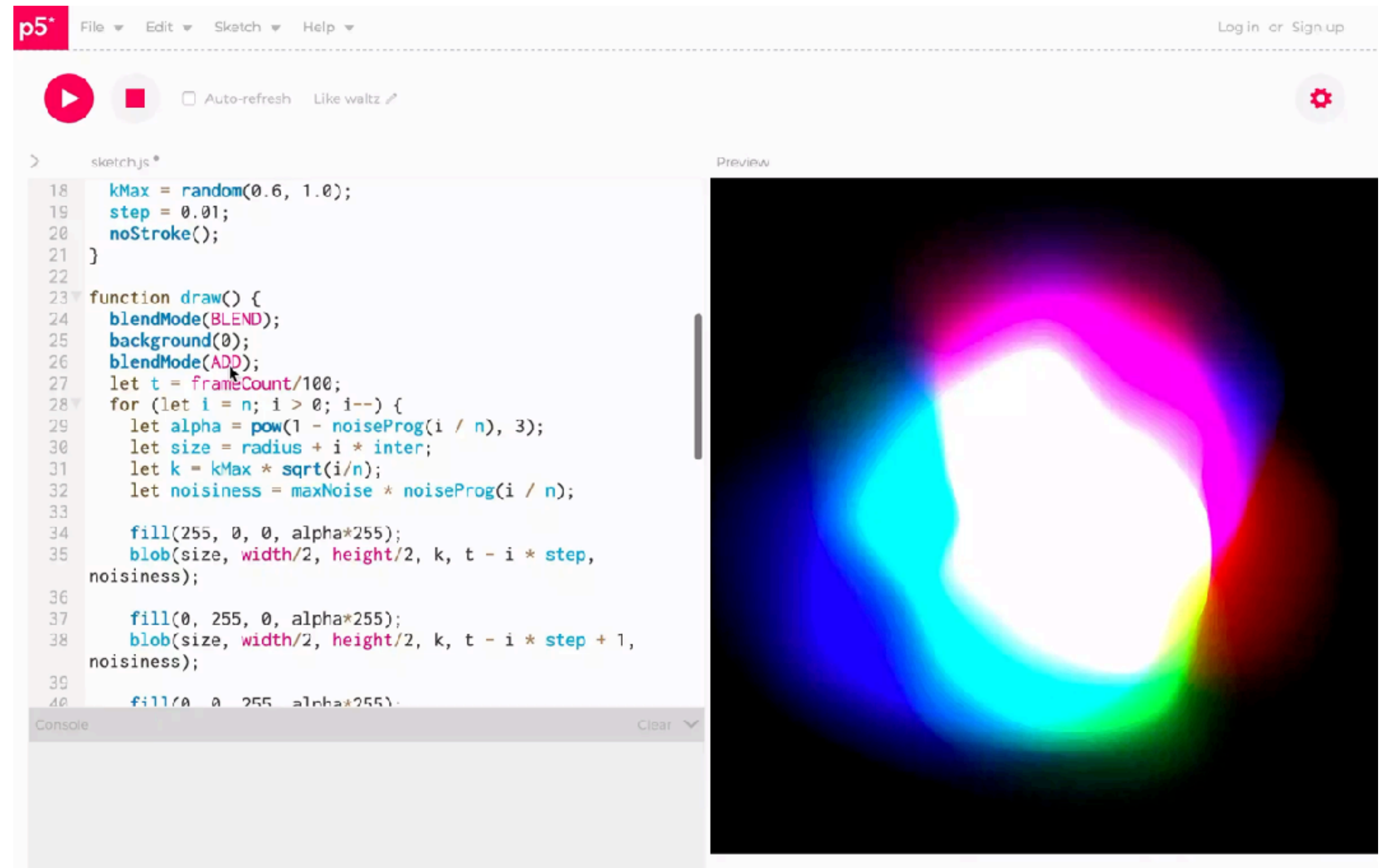
The screenshot shows the p5.js web editor interface. The top bar includes the p5.js logo, a menu (File, Edit, Sketch, Help), and a 'Log in or Sign up' link. Below the menu are a play button, a stop button, and a checkbox for 'Auto-refresh'. The main area is split into a code editor on the left and a preview window on the right. The code editor shows the following JavaScript code:

```
1 // By Roni Kaufman
2
3 let kMax;
4 let step;
5 let n = 80; // number of blobs
6 let radius = 0; // diameter of the circle
7 let inter = 0.1; // difference between the sizes of two
  blobs
8 let maxNoise = 500;
9
10 let noiseProg = (x) => (x);
11
12 function setup() {
13   createCanvas(600, 600);
14   //colorMode(HSB, 1);
15   angleMode(DEGREES);
16   noFill();
17   //noLoop();
18   kMax = random(0.6, 1.0);
19   step = 0.01;
20   noStroke();
21 }
22
```

The preview window on the right shows a colorful, glowing ring or blob shape on a black background. The ring is composed of many small, overlapping circles in various colors, creating a vibrant, multi-colored effect.

Accessible to new programmers, but different than manually making art

- Language designed in response to a frustration of the cultural elitism of programming
- Learning through editing lots of community made examples and tutorials, or in person workshops



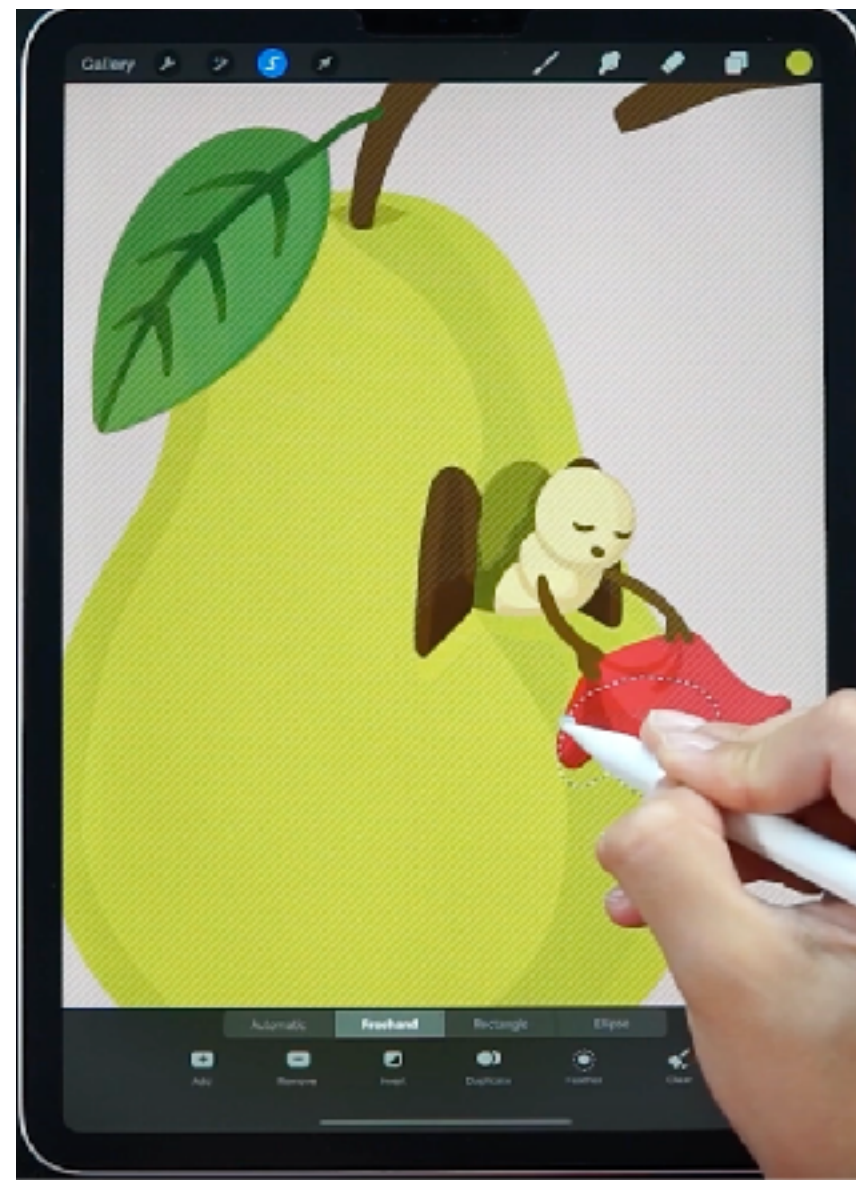
The screenshot shows the p5.js web editor interface. The top navigation bar includes 'p5*', 'File', 'Edit', 'Sketch', 'Help', and 'Login or Sign up'. Below the navigation bar are control buttons: a play button, a stop button, 'Auto-refresh', and 'Like waltz'. The main area is split into two panes: 'sketch.js*' on the left containing code, and 'Preview' on the right showing a colorful, abstract visualization.

```
18 kMax = random(0.6, 1.0);
19 step = 0.01;
20 noStroke();
21 }
22
23 function draw() {
24   blendMode(BLEND);
25   background(0);
26   blendMode(ADD);
27   let t = frameCount/100;
28   for (let i = n; i > 0; i--) {
29     let alpha = pow(1 - noiseProg(i / n), 3);
30     let size = radius + i * inter;
31     let k = kMax * sqrt(i/n);
32     let noisiness = maxNoise * noiseProg(i / n);
33
34     fill(255, 0, 0, alpha*255);
35     blob(size, width/2, height/2, k, t - i * step,
noisiness);
36
37     fill(0, 255, 0, alpha*255);
38     blob(size, width/2, height/2, k, t - i * step + 1,
noisiness);
39
40     fill(0, 0, 255, alpha*255);
```

The preview window displays a colorful, abstract visualization consisting of a central white shape surrounded by a thick, multi-colored ring. The colors transition from cyan and blue on the left to magenta and red on the right, with a greenish-yellow area at the bottom. The background is black.



Artwork



Canvas

- Directly manipulate the output
- Immediately observe how actions result in changes
- Allows for open-ended exploration



Artwork



Canvas

- Manipulate abstract symbols (code)
- Programming and execution are separate, unclear which pixel is caused by which line of code
- Requires more linear structure and building blocks before exploring

Your turn: p5.js studio



Home

Hello!

Editor

Download

Donate

Get Started

Reference

Libraries

Learn

Teach

Examples

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

Using the metaphor of a sketch, p5.js has a full set of drawing functionality. However, you're not limited to your drawing canvas. You can think of your whole browser page as your sketch, including HTML5 objects for text, input, video, webcam, and sound.

Start creating with the p5 Editor!

Contribute

Books

Community

Showcase

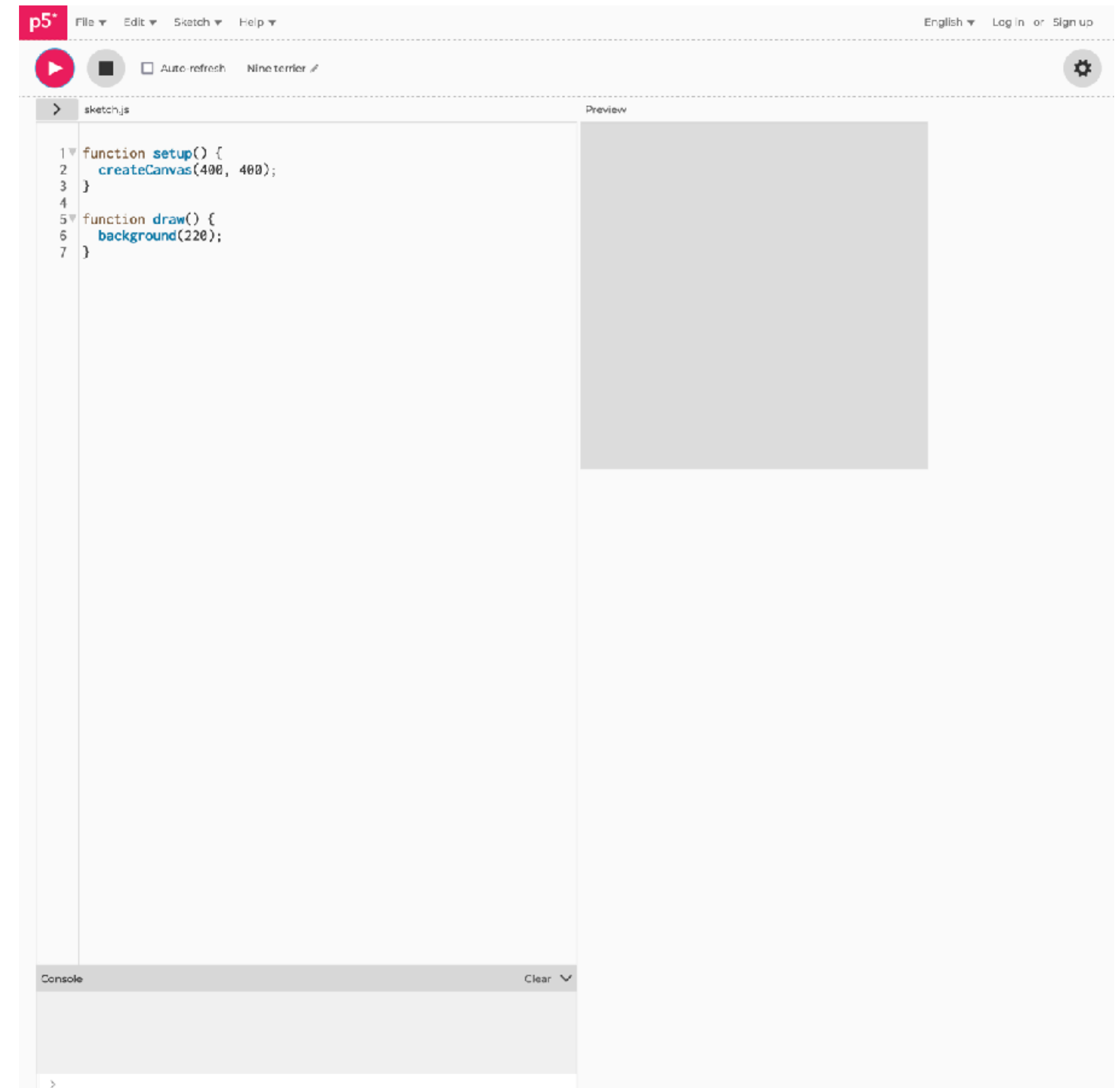
Community

We are a community of, and in solidarity with, people from every gender identity and expression, sexual orientation, race, ethnicity, language, neuro-type, size, disability, class, religion, culture, subculture, political opinion, age, skill level, occupation, and background. We acknowledge that not everyone has the time, financial means, or capacity to actively participate, but we recognize and encourage involvement of all kinds. We facilitate and foster access and empowerment. We are all learners.

p5.js is an interpretation of [Processing](#) for today's web. We hold events and operate with support from the [Processing Foundation](#).

Learn more about [our community](#).

p5js.org

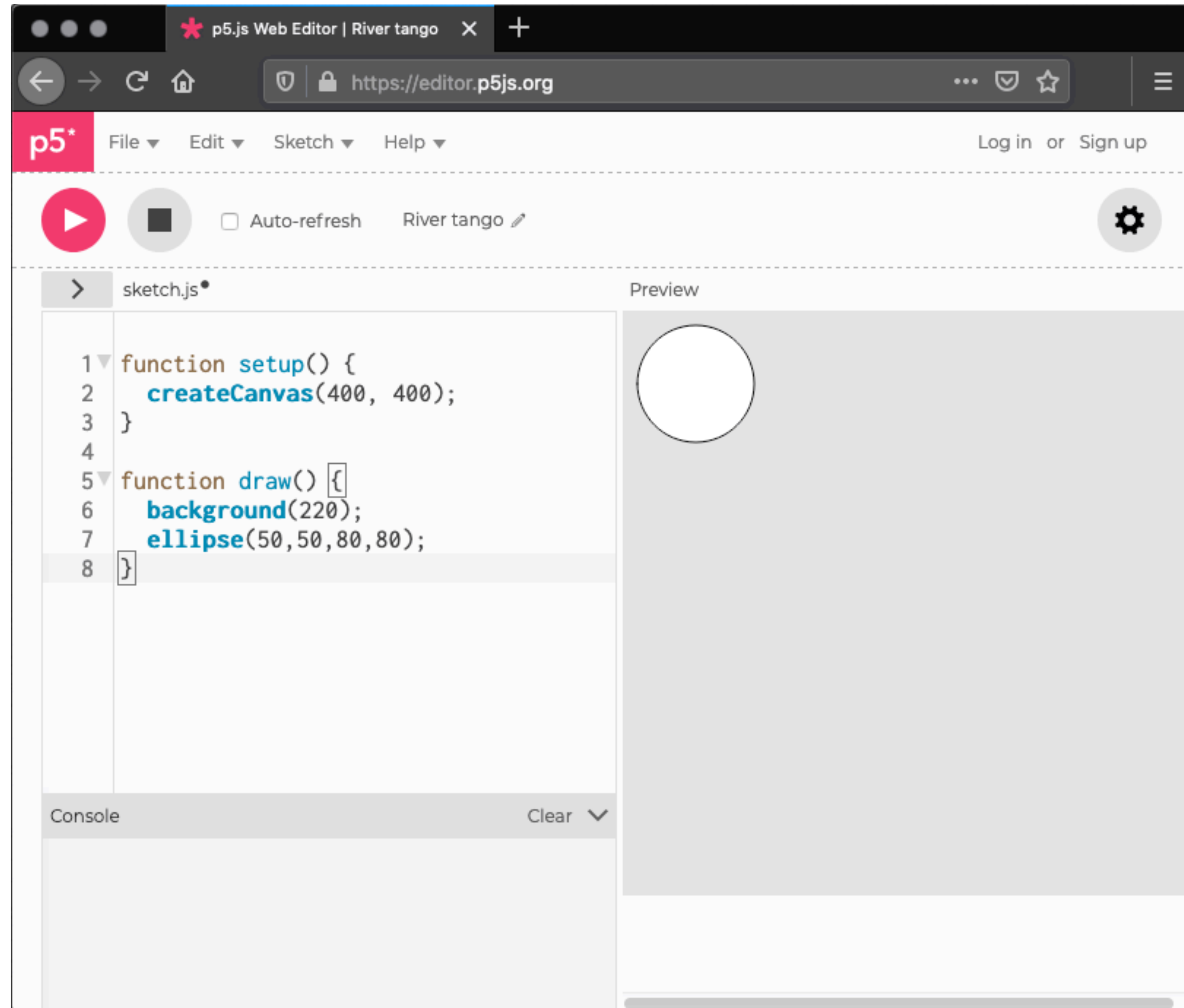


editor.p5js.org

Javascript syntax

setup() called
once at the
beginning

draw() called in a
loop



The screenshot shows the p5.js Web Editor interface. The browser address bar displays `https://editor.p5js.org`. The editor's top navigation bar includes the p5.js logo, menu items (File, Edit, Sketch, Help), and links for "Log in" or "Sign up". Below the navigation bar, there are control buttons: a play button, a stop button, an "Auto-refresh" checkbox, and the user name "River tango" with an edit icon. A settings gear icon is also present.

The main workspace is divided into two panels. The left panel, titled "sketch.js", contains the following JavaScript code:

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7   ellipse(50, 50, 80, 80);  
8 }
```

The right panel, titled "Preview", shows a white circle on a light gray background, which is the visual output of the code in the left panel.

At the bottom of the editor, there is a "Console" panel with a "Clear" button and a dropdown arrow.

Useful functions

background(color)

Examples

```
Press Shift-Space to insert tab. edit reset copy
// A grayscale integer value.
background(51);
describe('A canvas with a dark charcoal gray background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// A grayscale integer value and an alpha value.
background(51, 0.4);
describe('A canvas with a transparent gray background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// R, G & B integer values.
background(255, 204, 0);
describe('A canvas with a yellow background.');
```

```
Press Shift-Space to insert tab. edit reset copy
// H, S & B integer values.
colorMode(HSB);
background(255, 204, 100);
describe('A canvas with a royal blue background.');
```

color can be

- 1 argument: grayscale value (0-255)
- 2 arguments: grayscale value & opacity (0-1)
- 3 arguments: (red, green, blue) (0-255)
- 3 arguments: (hue, saturation, value)
- 1 argument: hex code '#00ff00'
- 1 argument: CSS named color 'magenta'
- and more!

fill(color)

applies to all shapes after

```
Press Shift-Space to insert tab. edit reset copy
// Six-digit hex RGB notation.
fill('#A251FA');
square(20, 20, 60);
describe('A purple square with a black outline.');
```

what kinds of shapes?

line(x1, y1, x2, y2)

ellipse(x, y, w, [h])

rect(x, y, w, [h])

...

Read the reference docs!!!

[Home](#)[Editor](#)[Download](#)[Donate](#)[Get Started](#)[Reference](#)[Libraries](#)[Learn](#)[Teach](#)[Examples](#)[Contribute](#)[Books](#)[Community](#)[Showcase](#)

Reference

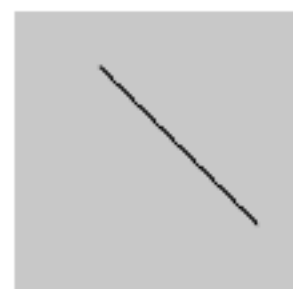
line()

Description

Draws a line, a straight path between two points. Its default width is one pixel. The version of `line()` with four parameters draws the line in 2D. To color a line, use the `stroke()` function. To change its width, use the `strokeWeight()` function. A line can't be filled, so the `fill()` function won't affect the color of a line.

The version of `line()` with six parameters allows the line to be drawn in 3D space. Doing so requires adding the `WEBGL` argument to `createCanvas()`.

Examples



Press Shift-Space to insert tab.

[edit](#) [reset](#) [copy](#)

```
line(30, 20, 85, 75);
describe(
  'A black line on a gray canvas running from
  top-center to bottom-right.'
);
```

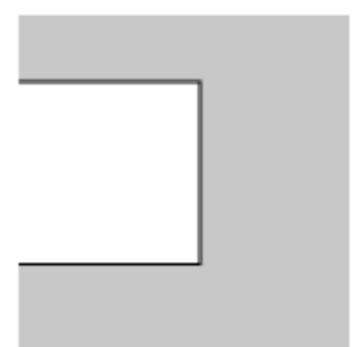

Useful functions

transformations stack & apply to objects after

translate(x, y)

rotate(angle) ...

Examples



Press Shift-Space to insert tab.

edit reset copy

```
translate(0, 20);  
rect(0, 0, 55, 55);
```

translate(20, 0)

translate(50, 0)



translate(70, 0)

same thing

<old settings>

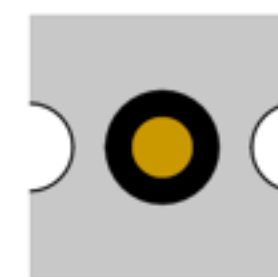
push()

<new settings>

pop()

<old settings>

Examples



Press Shift-Space to insert tab.

edit reset copy

```
ellipse(0, 50, 33, 33); // Left circle
```

```
push(); // Start a new drawing state
```

```
strokeWeight(10);
```

```
fill(204, 153, 0);
```

```
translate(50, 0);
```

```
ellipse(0, 50, 33, 33); // Middle circle
```

```
pop(); // Restore original state
```

```
ellipse(100, 50, 33, 33); // Right circle
```

Home

Editor

Download

Donate

Get Started

Reference

Libraries

Learn

Teach

Examples

Contribute

Books

Community

Showcase

Programming topics

Beyond the canvas

Creating and manipulating elements on the page beyond the canvas.

Coordinate System and Shapes

Drawing simple shapes and using the coordinate system.

Interactivity

Introduction to interactivity with the mouse and keyboard.

Color

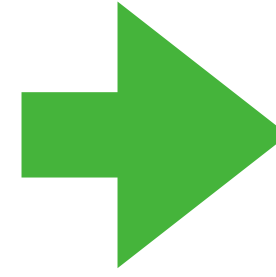
An introduction to digital color.

Curves

An introduction to the three types of curves in p5.js: arcs, spline curves, and Bézier curves.

Program Flow

Introduction to controlling program flow in p5.js.



Home

Editor

Download

Donate

Get Started

Reference

Libraries

Learn

Teach

Examples

Contribute

Books

Community

Showcase

Reference

Can't find what you're looking for? You may want to check out [p5.sound](#). You can also download an offline version of the reference.

3D

Data

IO

Shape

Color

Environment

Image

Structure

Constants

Events

Math

Transform

DOM

Foundation

Rendering

Typography

Learn

Teach

Examples

Environment Color

describe()

describeElement()

textOutput()

gridOutput()

print()

frameCount

deltaTime

focused

cursor()

frameRate()

getTargetFrameRate()

noCursor()

webglVersion

displayWidth

displayHeight

windowWidth

windowHeight

windowResized()

width

height

Creating & Reading

alpha()

blue()

brightness()

color()

green()

hue()

lerpColor()

lightness()

red()

saturation()

p5.Color

Setting

beginClip()

endClip()

clip()

background()

clear()

colorMode()

fill()

noFill()

noStroke()

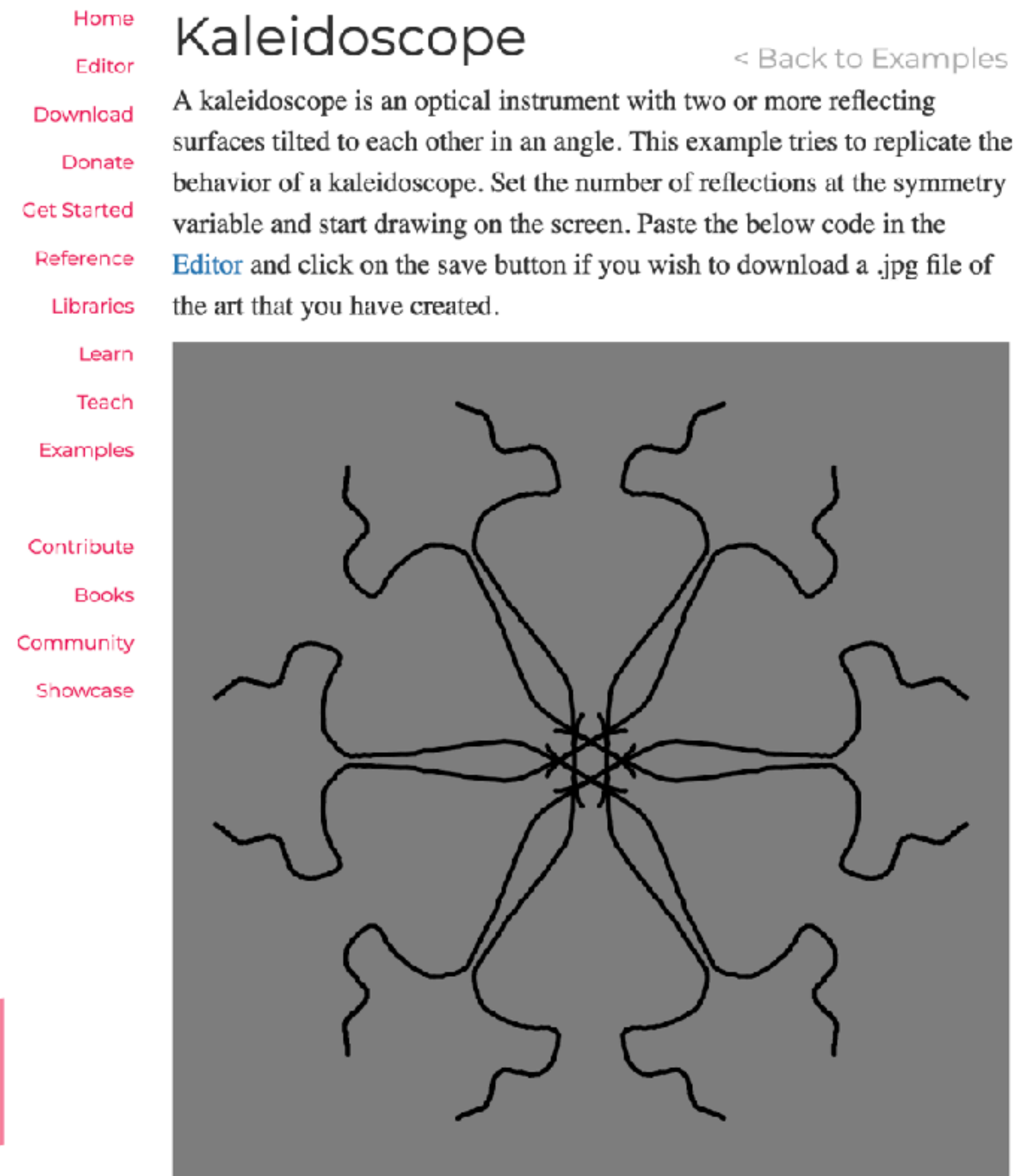
stroke()

erase()

noErase()

Your task: edit Kaleidoscope example

- Examples (sidebar) -> Interaction -> Kaleidoscope
- Paste example code into editor
- Make at least **3 meaningful lines of code changes** resulting in a **visually different** piece
- Learning goal: experiment with p5.js, just change random variables and see what happens, understand creative coding process
- Save and upload drawings on Canvas (p5.js art gallery assignment)



The screenshot shows the p5.js website's 'Kaleidoscope' example page. On the left is a vertical sidebar with navigation links: Home, Editor, Download, Donate, Get Started, Reference, Libraries, Learn, Teach, Examples, Contribute, Books, Community, and Showcase. The main content area has the title 'Kaleidoscope' and a '< Back to Examples' link. Below the title is a paragraph explaining that a kaleidoscope is an optical instrument with two or more reflecting surfaces tilted to each other in an angle, and that the example tries to replicate its behavior. It instructs the user to set the number of reflections at the symmetry variable and start drawing on the screen, then to paste the code in the Editor and click the save button to download a .jpg file. At the bottom of the page is a large, dark gray square containing a black line drawing of a complex, symmetrical, fractal-like pattern with multiple layers of self-similar, wavy shapes radiating from a central point.

Instructions

- Use the Reference and Learn links to understand the p5.js API and brainstorm changes
- Pair program to change the kaleidoscope example
- Make at least **3 meaningful lines of code changes** resulting in a **visually different** piece
- Save and upload drawings on Canvas (p5.js art gallery assignment)
- If you're done early, type a short reflection: How did this experience differ from other kinds of coding? What was challenging about being expressive?

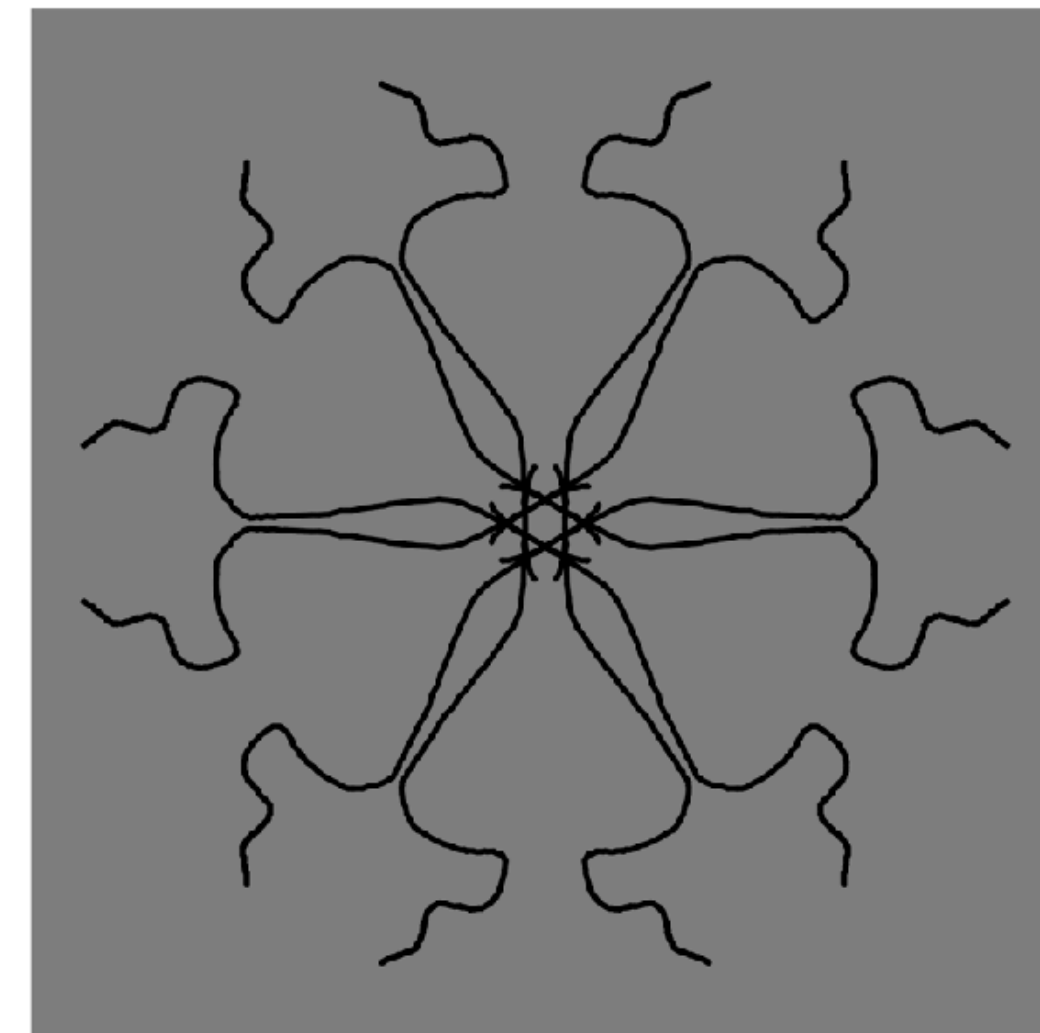
p5.js

Home
Editor
Download
Donate
Get Started
Reference
Libraries
Learn
Teach
Examples
Contribute
Books
Community
Showcase

Kaleidoscope

[< Back to Examples](#)

A kaleidoscope is an optical instrument with two or more reflecting surfaces tilted to each other in an angle. This example tries to replicate the behavior of a kaleidoscope. Set the number of reflections at the symmetry variable and start drawing on the screen. Paste the below code in the [Editor](#) and click on the save button if you wish to download a .jpg file of the art that you have created.



This assignment does not count toward the final grade.

p5.js art gallery

Published

Edit

⋮

Upload your image here (one person per group is fine).

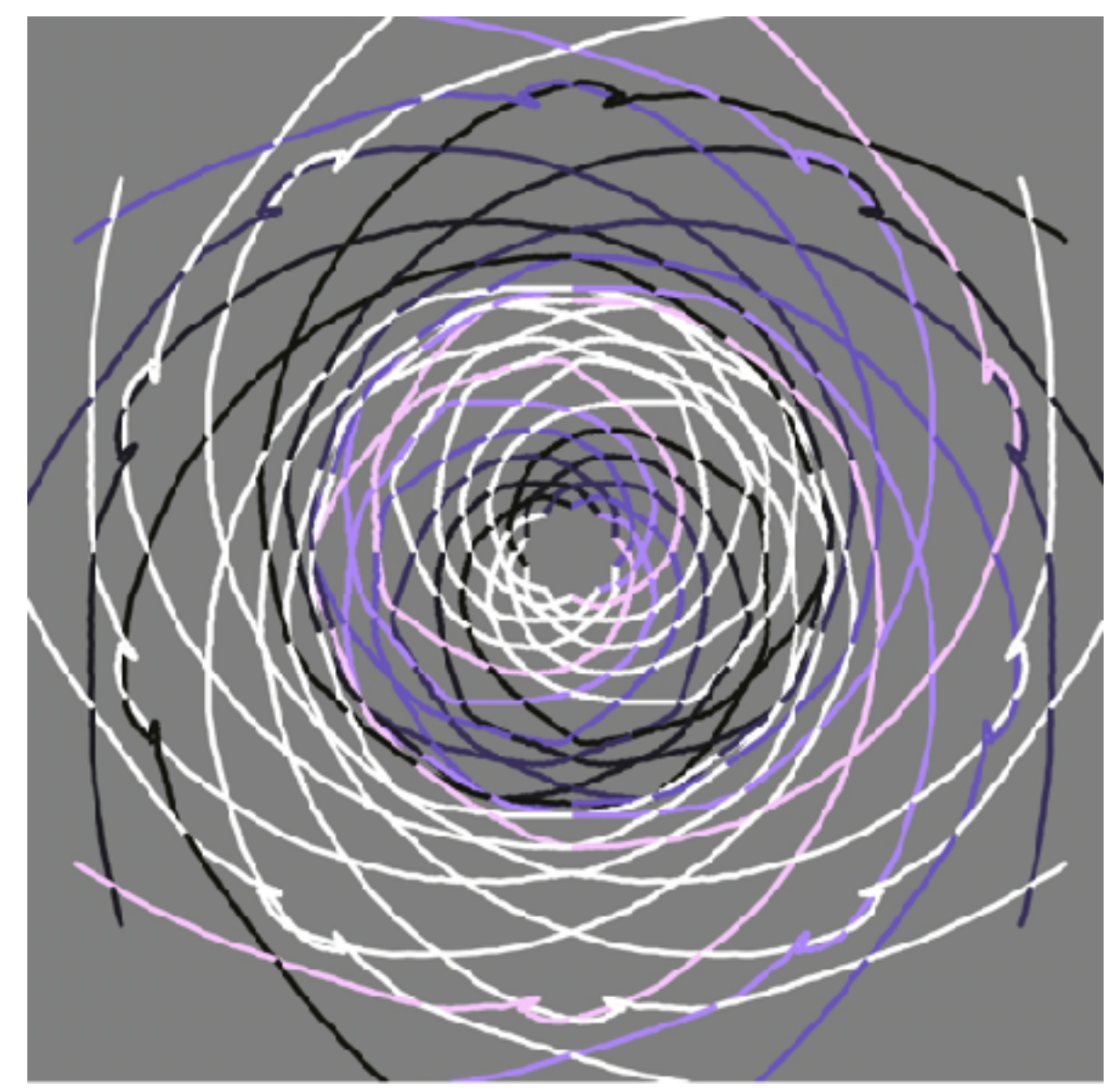
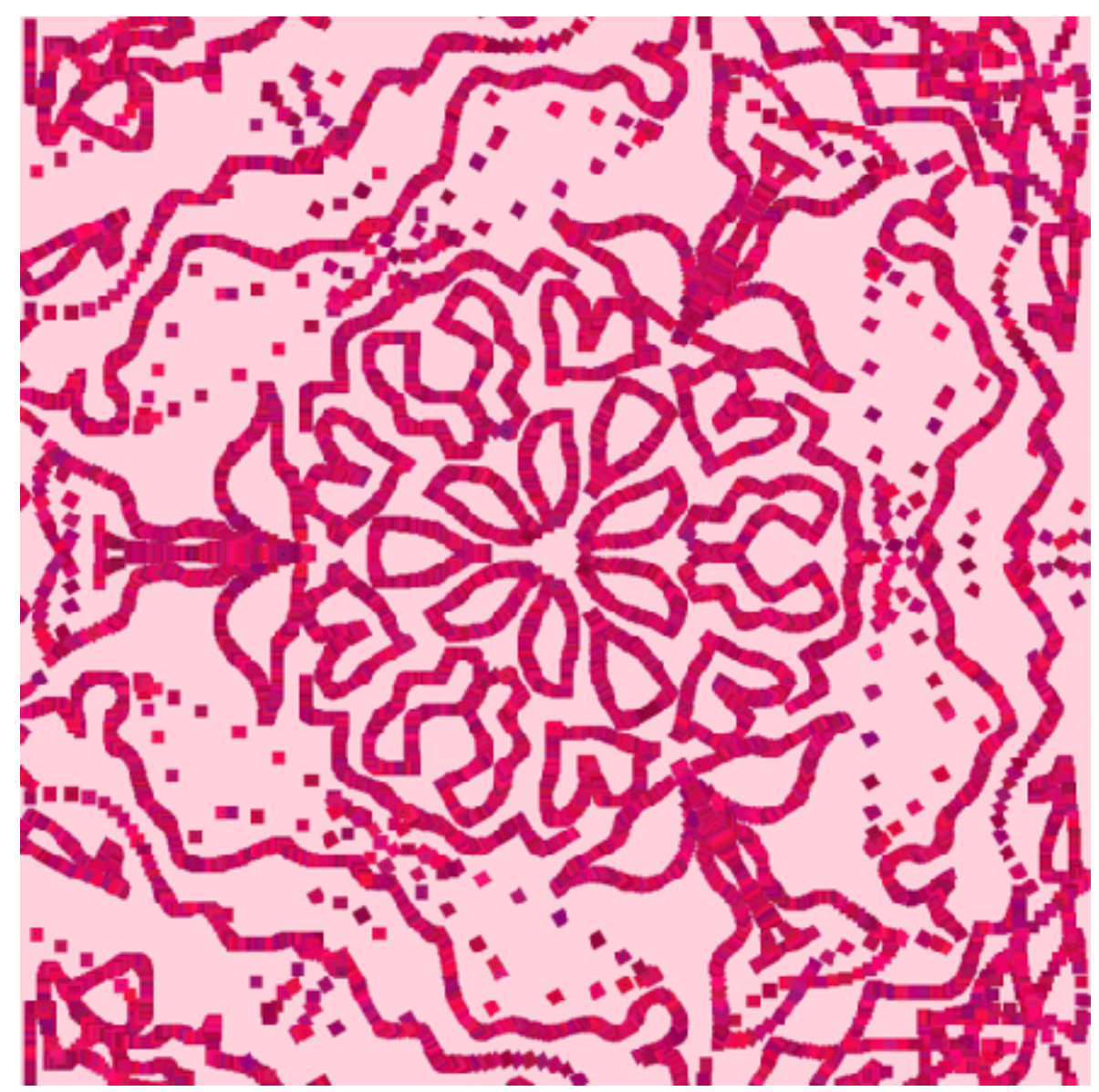
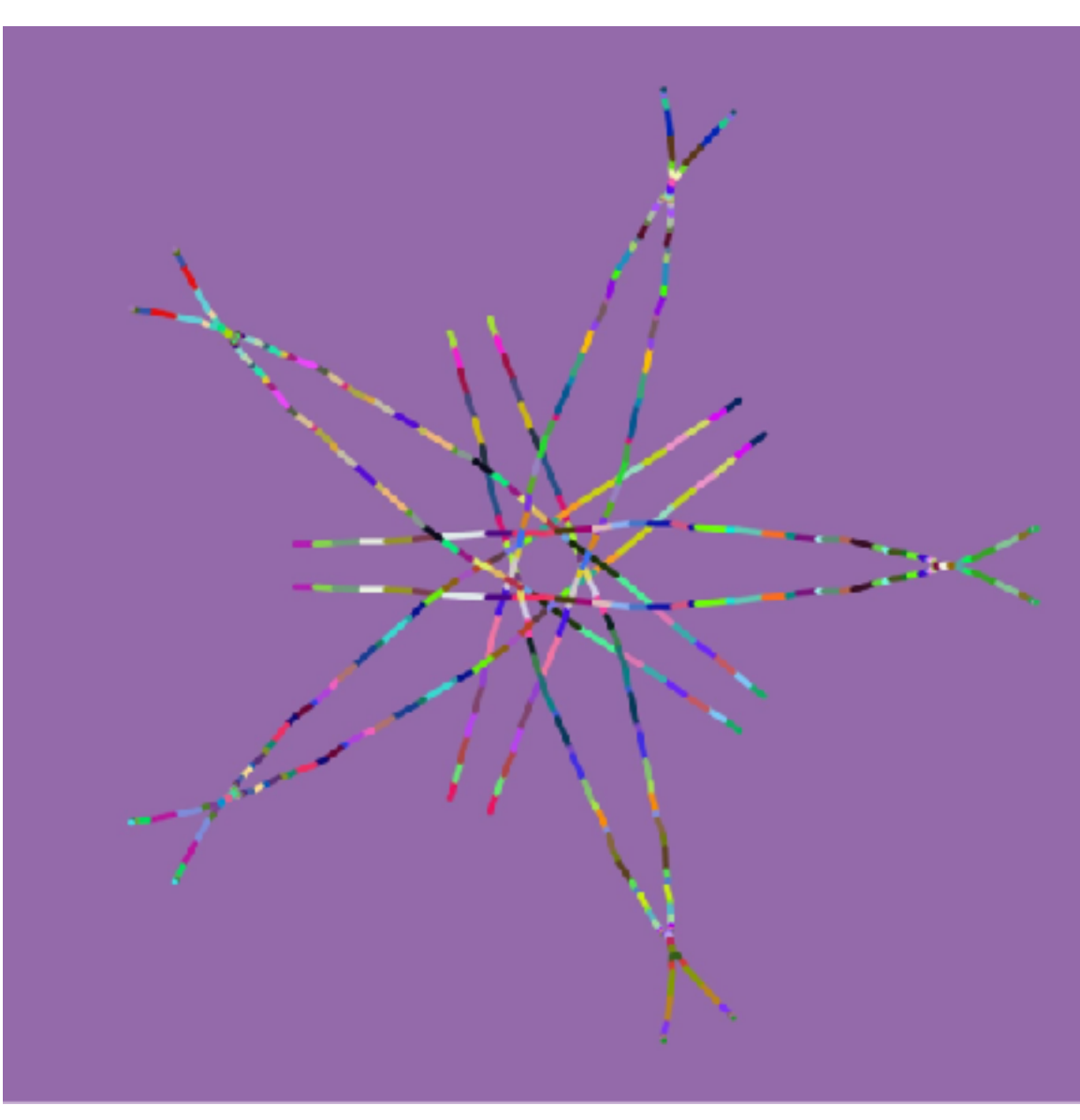
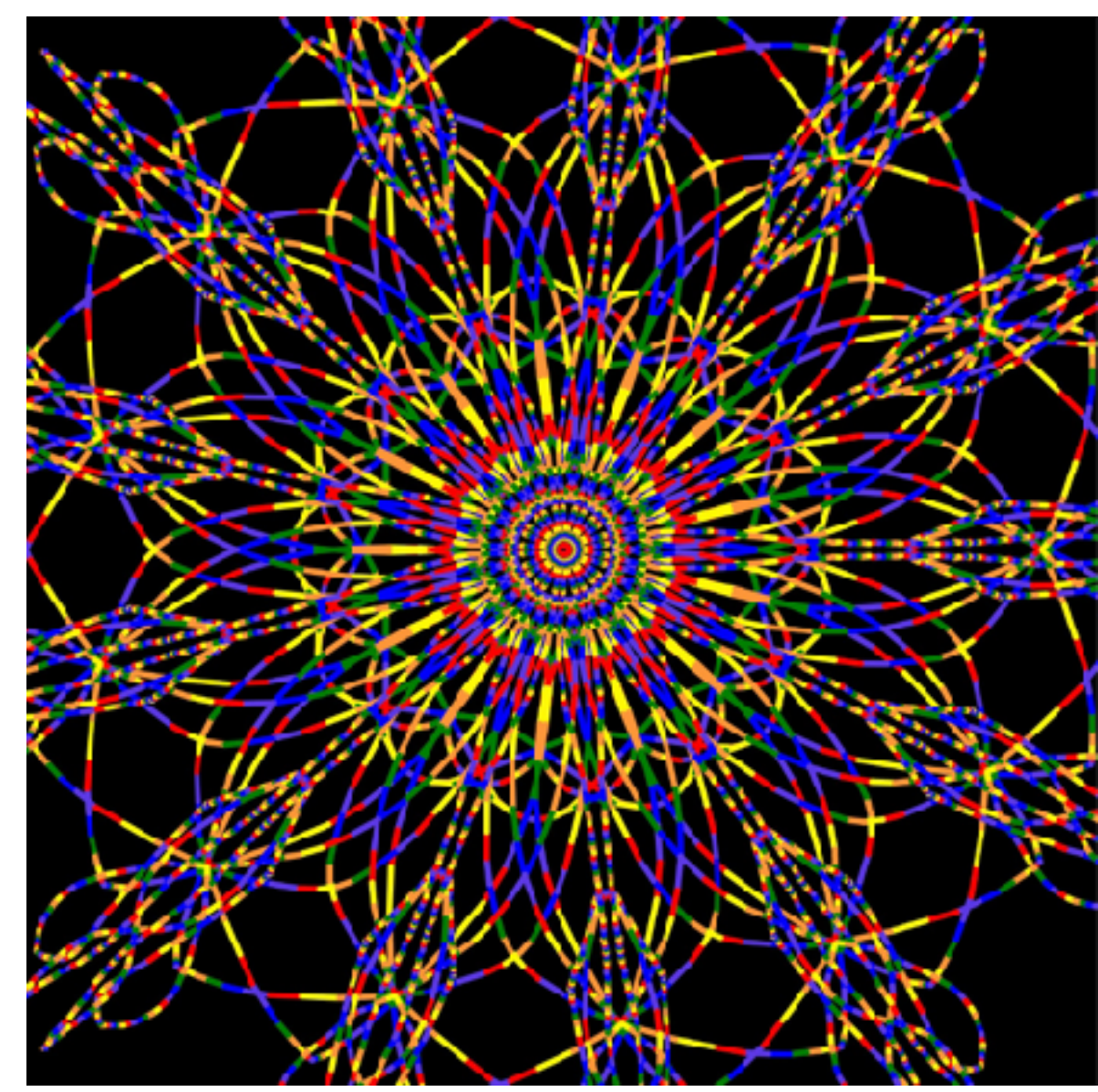
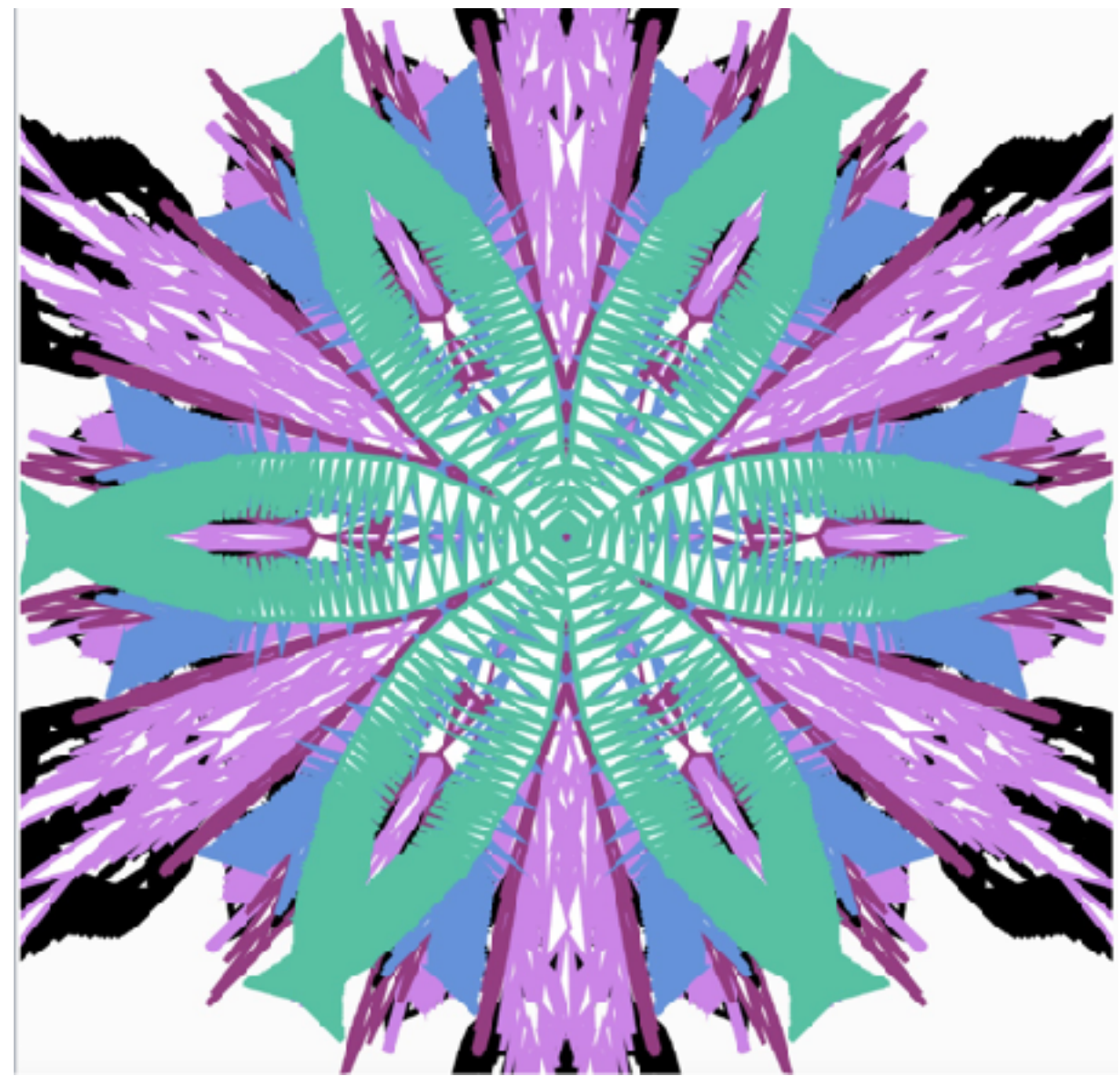
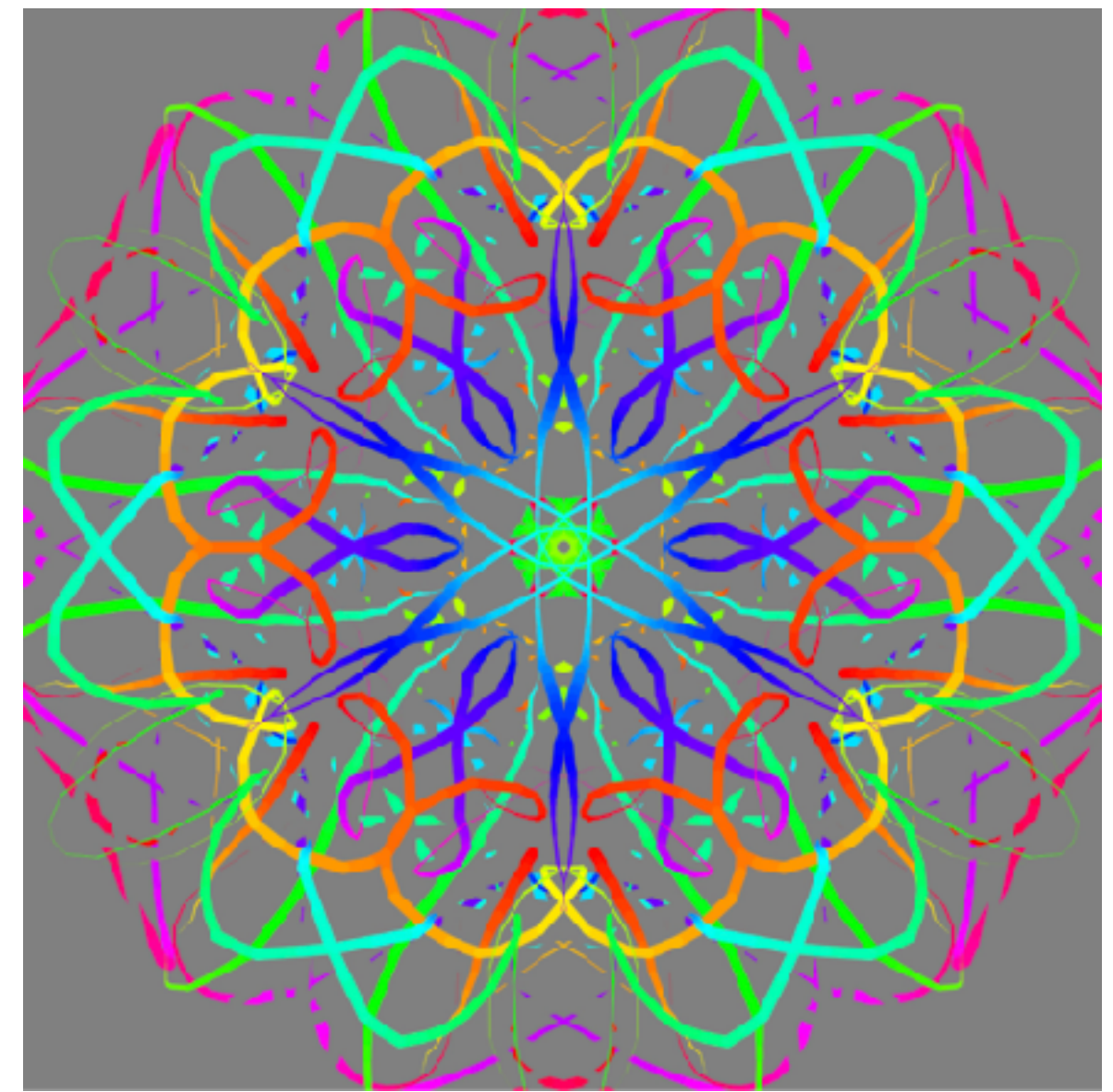
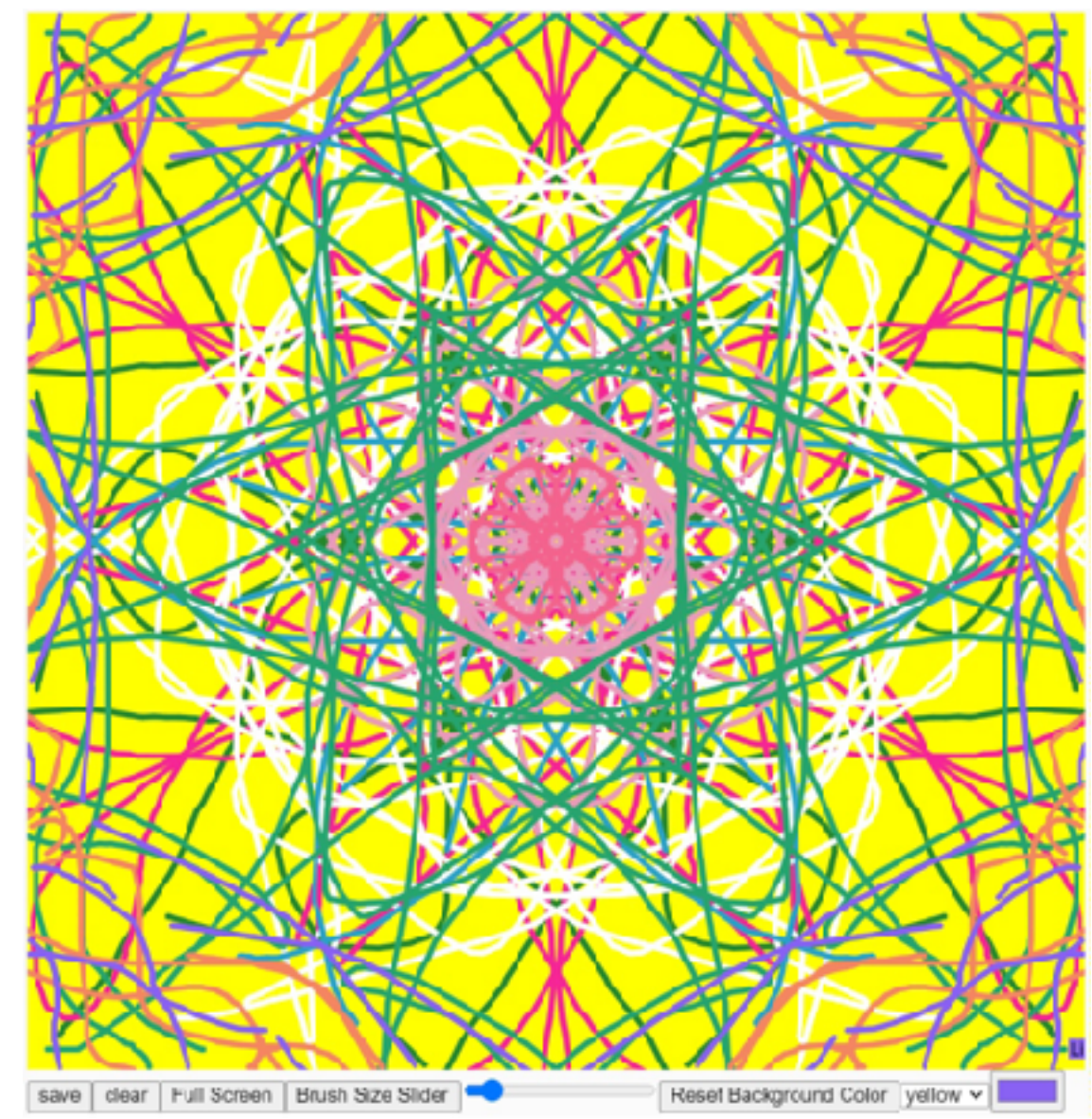
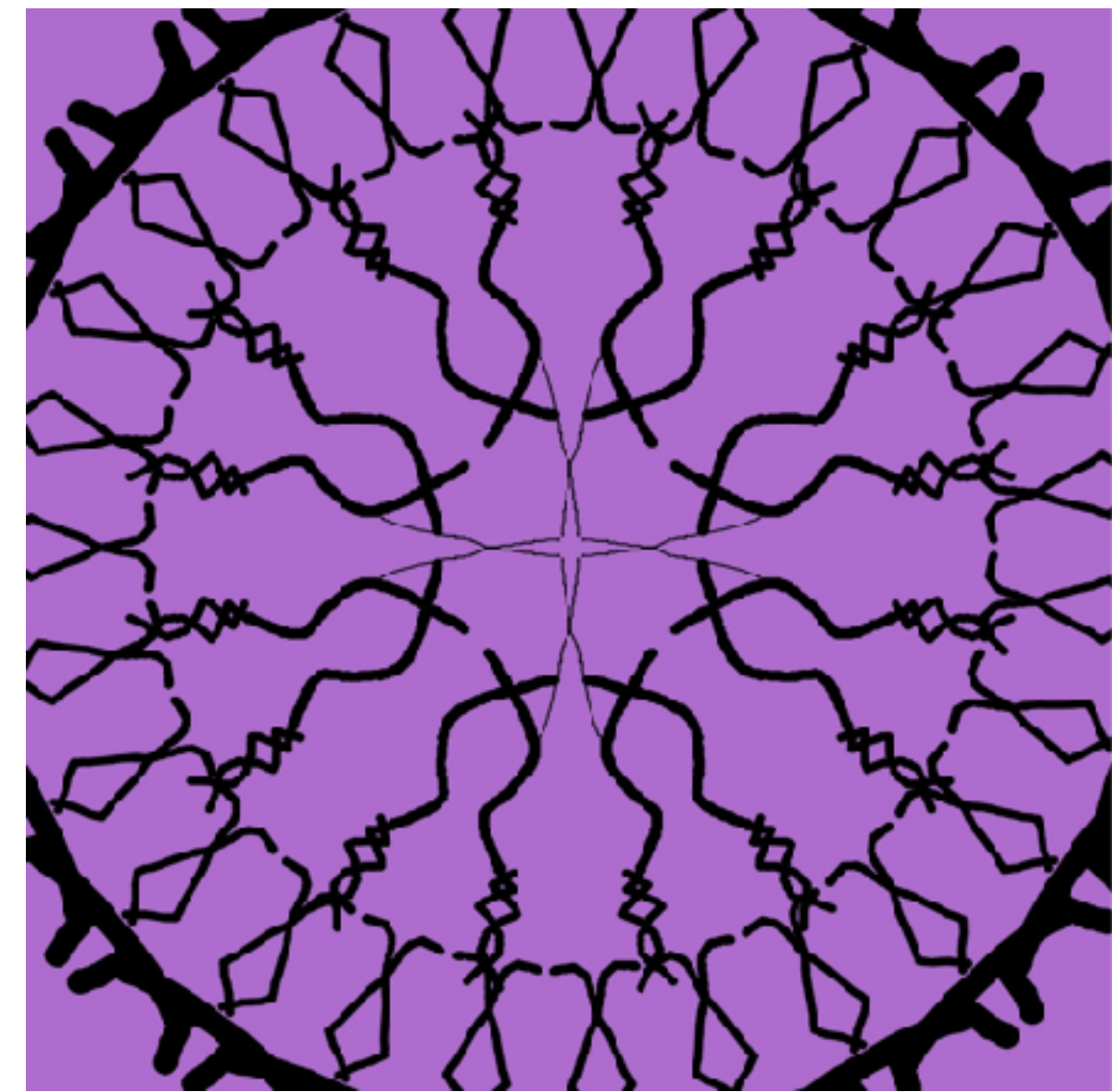
Points 0

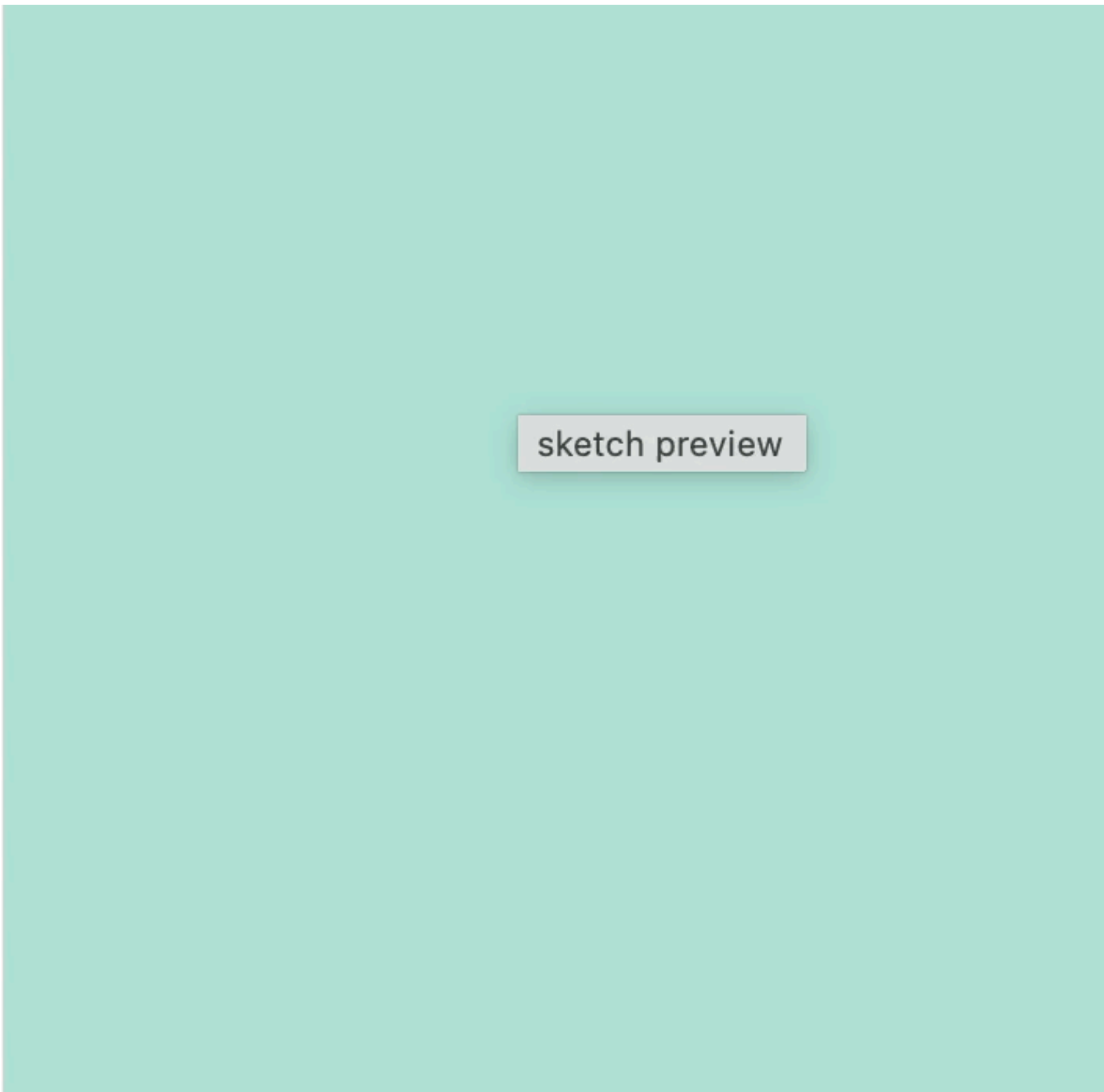
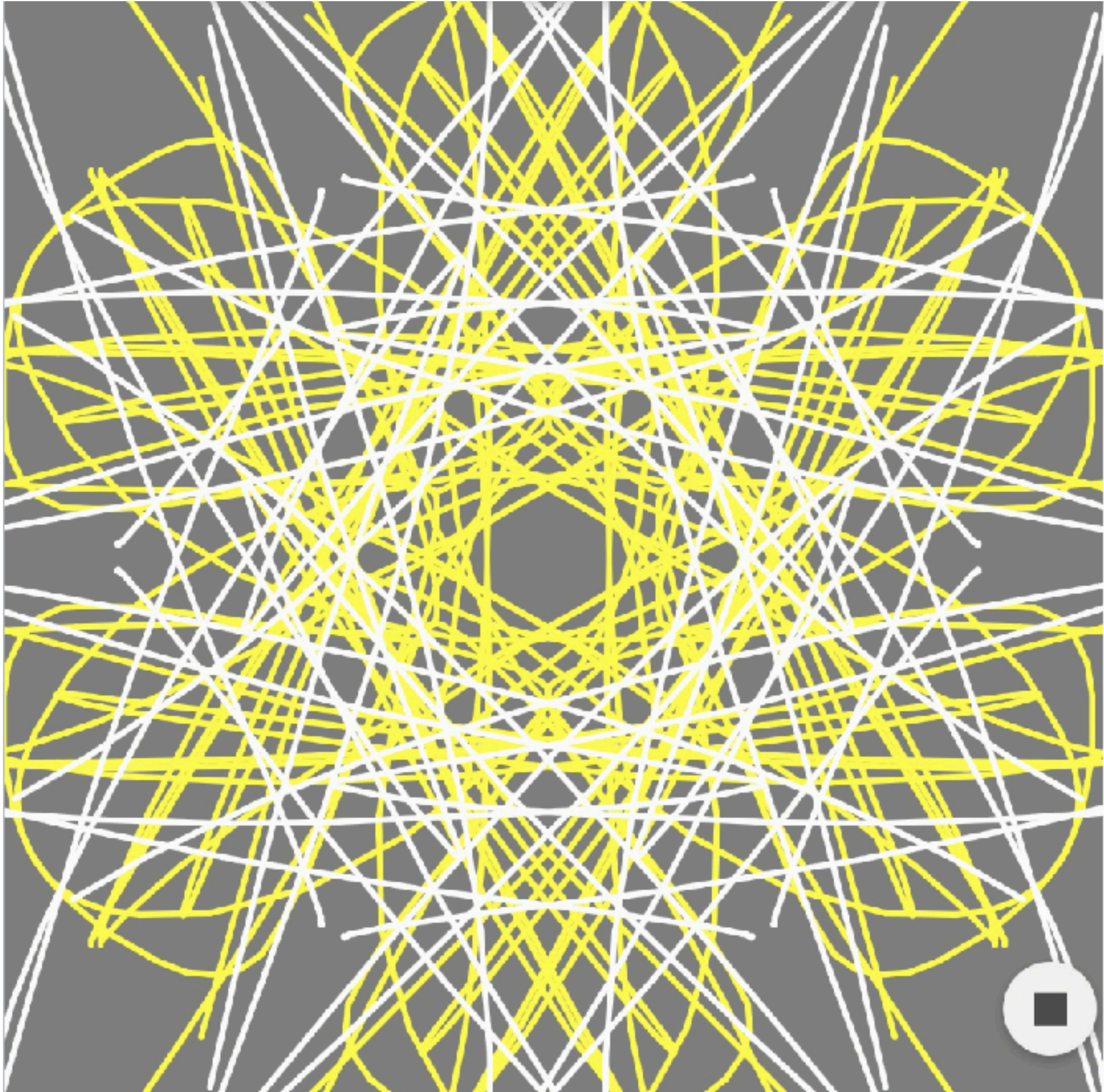
Submitting a file upload



How did your experience of using p5.js and creative coding differ from your usual CS experience? What were some challenges of being expressive?

Join at
slido.com
#1817





Class 7 recap

- TODOs:
 - By **Thursday's** class:
 - Bring what you have for P1 to class - it's project work time and OH (a good chance to get feedback!)
 - Next week
 - My **Monday** OH moved to 5:45-6:30pm, or by appointment after 7pm
 - Tues: P1 presentations
 - Thurs: PM3 (press fit kit) art walk & digital fabrication design tools seminar