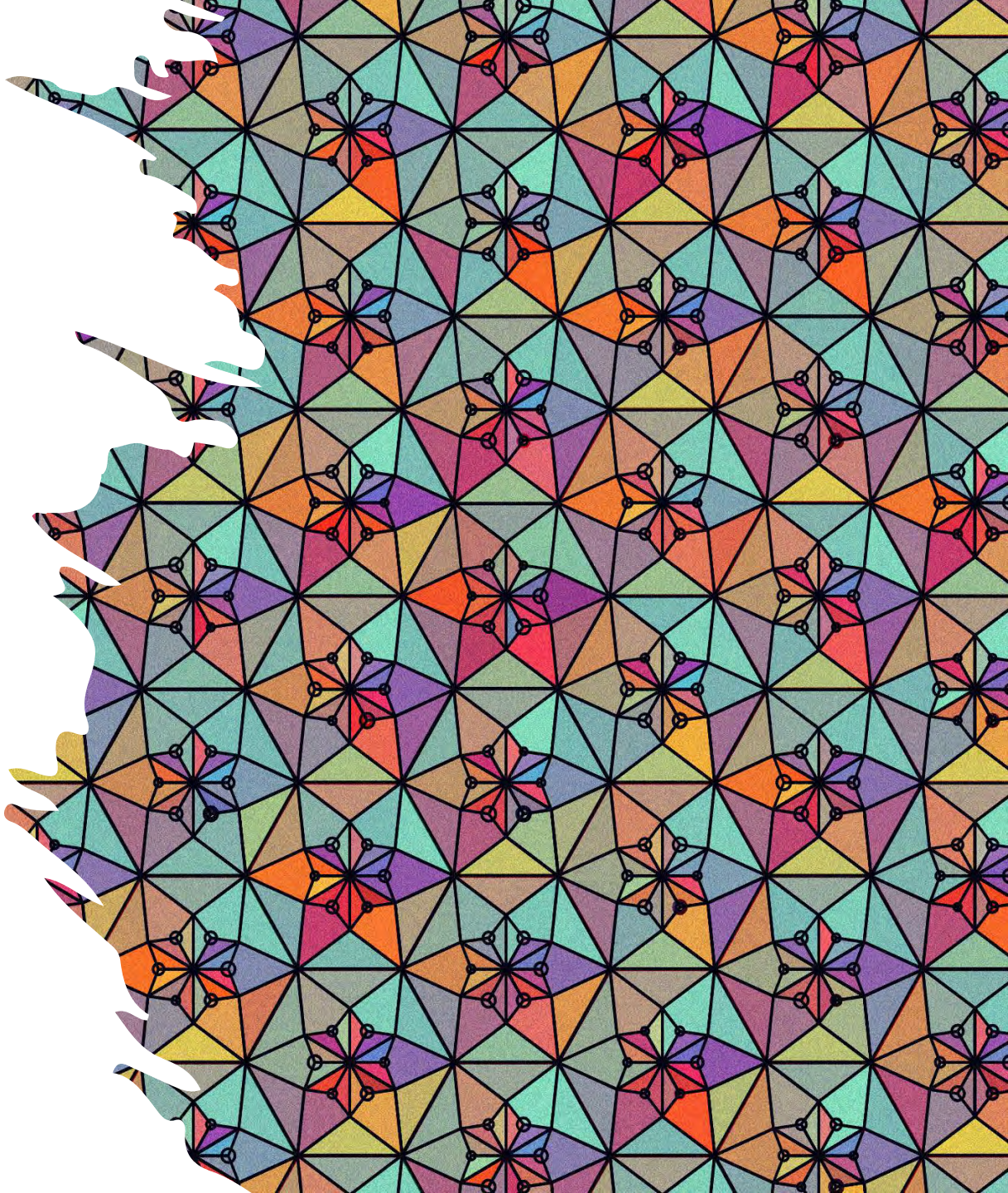
Infinite Patterns in the Digital Canvas

Unleashing creativity with **JavaScript** in Algorithmic Arts

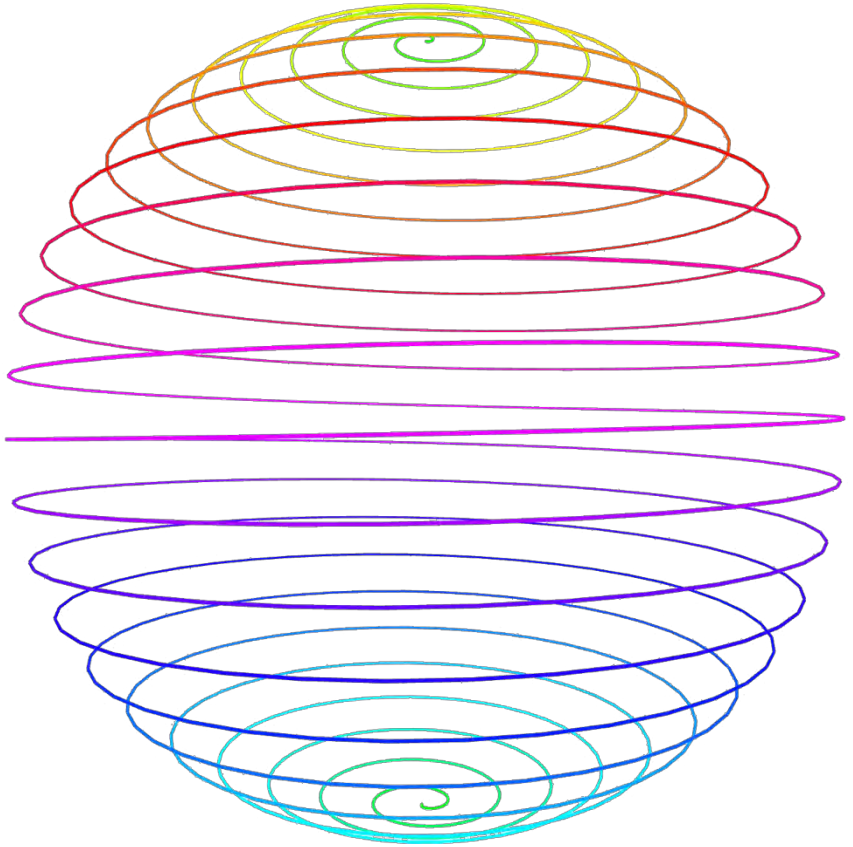
frani.be 🇨🇱



frani.be

frani.be

Table of Contents



Personal Bio

Francisca Beatriz Medina Concha (frani.be)

My Roles

UX + Front-End Leader

Data Visualization Team, LATAM Airlines

Speaker

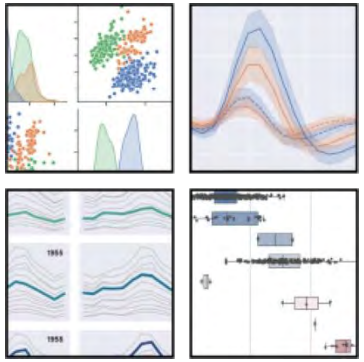
JSConf Chile

Volunteer

WTM Chile

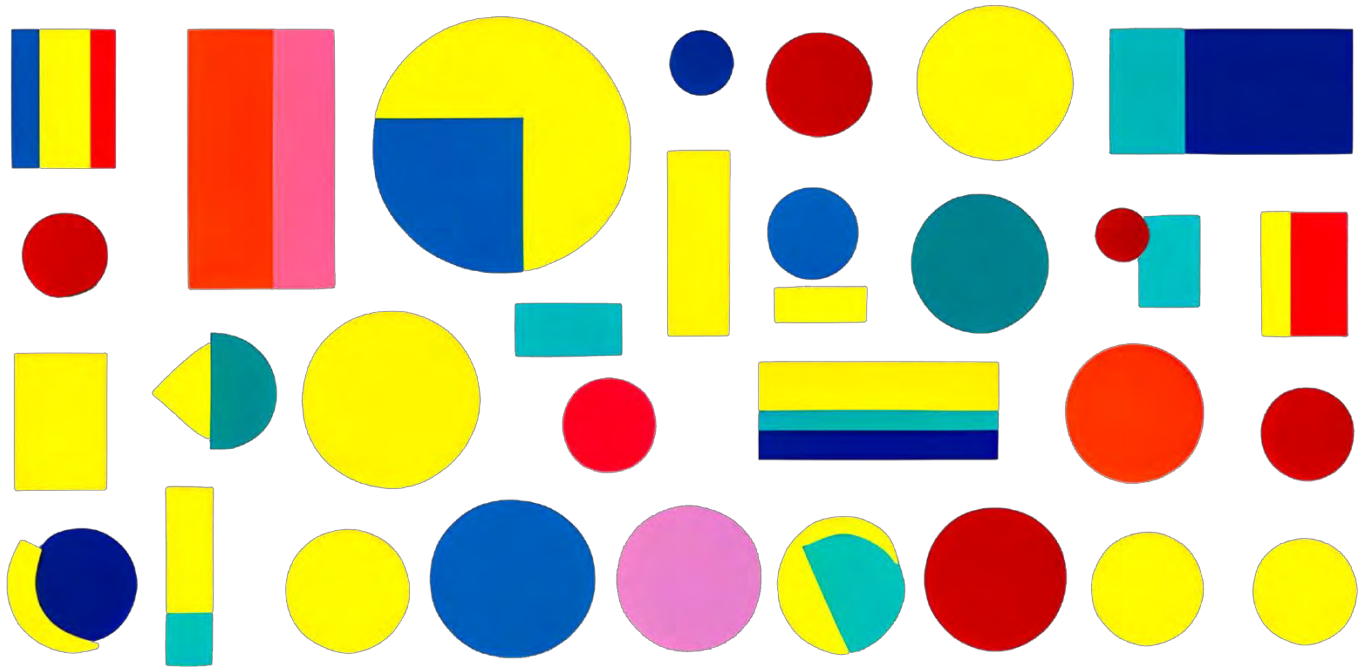
frani.be

Who am I?



frani.be

Personal Motivations...



Algorithmic Art Introduction

Although the artist sets the rules, the exact result can be **unpredictable** and **surprisingly complex**.

frani.be

What is Art?

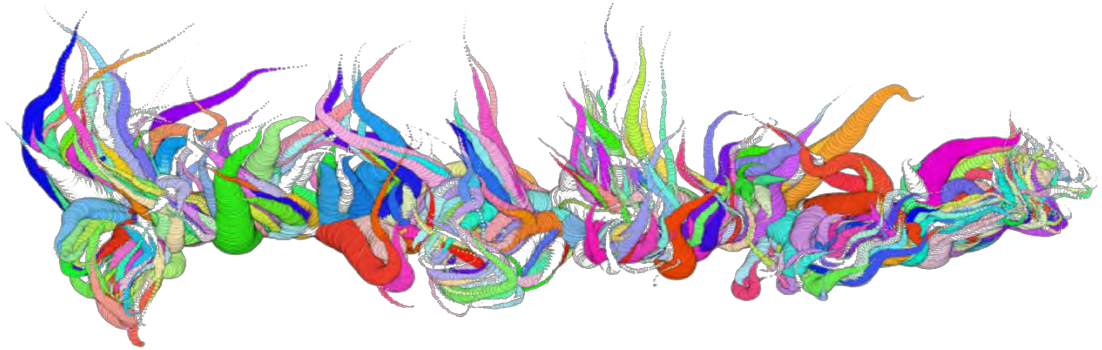
What is Art?

- Art can be understood in many ways today.
- Defining art exclusively has become increasingly difficult.

Algorithmic Art

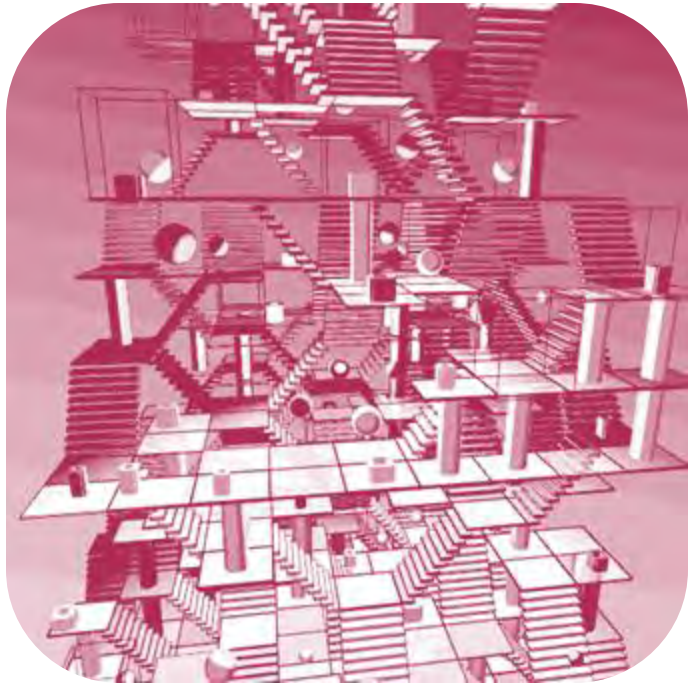
Algorithmic art represents a unique form of artistic expression where technology and programming play a fundamental role in the creation of the work.

This type of art is notable for its ability to transform algorithms and data into visual and sensory experiences, often challenging our traditional perceptions of art.



frani.be

Involves Algorithms Based Processes



```
let cw, ch
let building
let gl, test, theShader
let isOrtho, bgGradient, cf
function setup() {
  cw = 1000
  ch = 1000
  createCanvas(cw, ch, WEBGL)
  pixelDensity(1)
  createBgGradient()
  background(255)

  building = new Building()
  theShader = createFilterShader(frag)
  cf = floor(random(1000))
}

function draw() {
  if (isOrtho) {
    ortho()
  } else {
    perspective()
  }

  clear()
  background(255)
  push()
  translate(0, 0, -ch / 2)
  scale(2, 2, 1)
  image(bgGradient, -cw / 2, -ch / 2, cw, ch)
  pop()
```

The “Algorist”

An algorist is an artist who creates art with algorithms...

```
if (creation && object of art && algorithm && one's own algorithm) {
      include * an algorist *
} elseif (!creation || !object of art || !algorithm || !one's own algorithm) {
      exclude * not an algorist *
}
```

The “Algorist”

```
if (creation && object of art && algorithm && one's own algorithm) {
      include * an algorist *
} elseif (!creation || !object of art || !algorithm || !one's own algorithm) {
      exclude * not an algorist *
}
```

Jean-Pierre Hébert (September 1995)


JS

p5.js Library

JavaScript library with a focus on creating graphics and interactive web experiences

p5.js Introduction

p5.js is a JavaScript library created by Lauren McCarthy.

Its goal is to make coding accessible to artists, designers, educators and beginners.

It is inspired by Processing, a programming platform for visual art.

p5.js History

2001	Processing was developed by Casey Reas (artist) and Ben Fry (data scientist)
2008	Processing.js was developed by John Resig (programmer)
2013	p5.js was developed by Lauren McCarthy (artist and programmer)
2014-2015	Some Java and Procesing updates made it difficult for Processing.js to remain in use
2016	Processing.js stopped active development
2018	p5.js online editor launched
2020	Both Processing and p5.js continued to be actively used and developed.

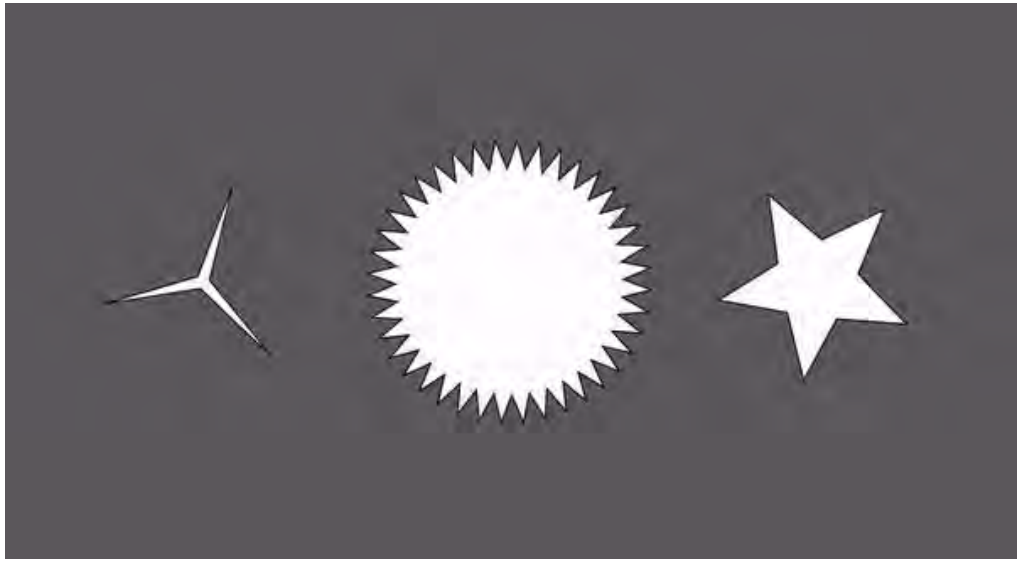
Processing vs p5.js

Exploring p5.js

- function setup()
- function draw()
- Canva (HTML tag)
- Colors
- Figures

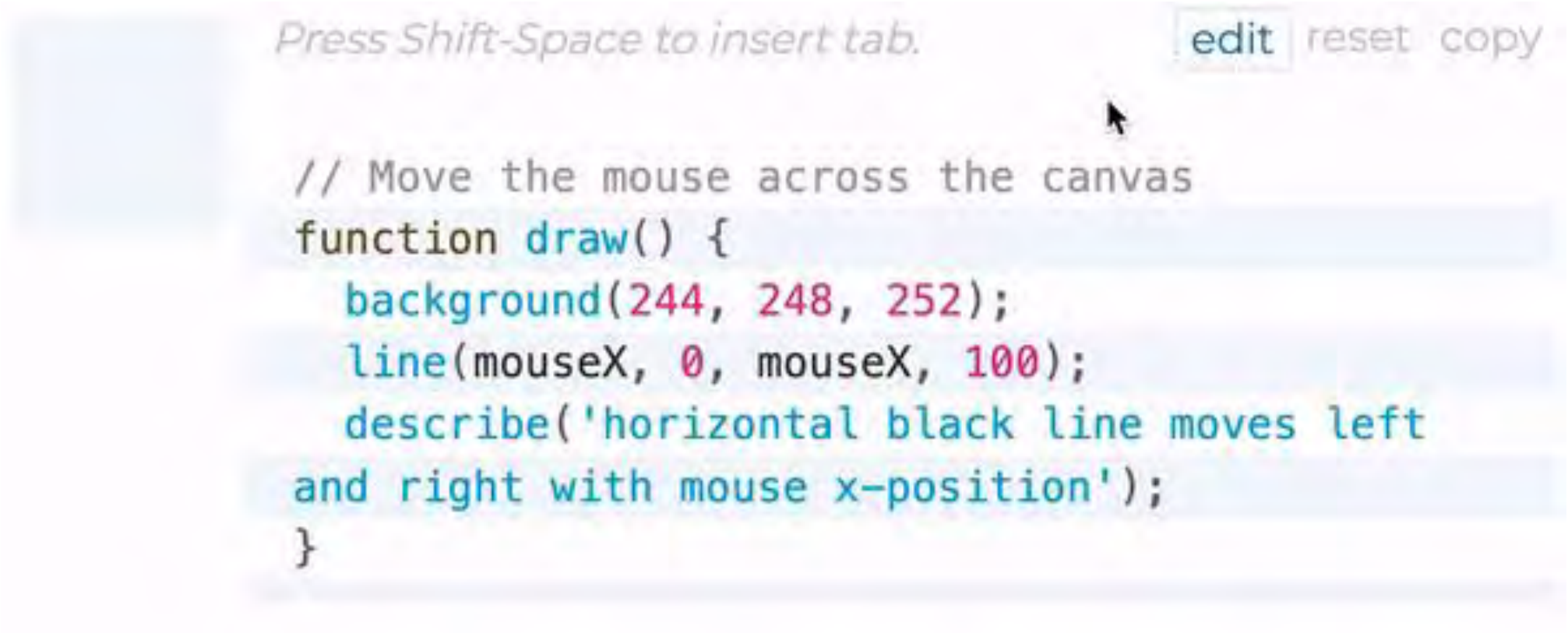
Exploring p5.js

```
function setup() {
  createCanvas(720, 400);
}

function draw() {
  background(102);

  push();
  translate(width * 0.2, height * 0.5);
  rotate(frameCount / 200.0);
  star(0, 0, 5, 70, 3);
  pop();

  push();
  translate(width * 0.5, height * 0.5);
  rotate(frameCount / 50.0);
  star(0, 0, 80, 100, 40);
  pop();

  push();
  translate(width * 0.8, height * 0.5);
  rotate(frameCount / -100.0);
  star(0, 0, 30, 70, 5);
  pop();
}

function star(x, y, radius1, radius2, npoints) {
  let angle = TWO_PI / npoints;
  let halfAngle = angle / 2.0;
  beginShape();
  for (let a = 0; a < TWO_PI; a += angle) {
    let sx = x + cos(a) * radius2;
    let sy = y + sin(a) * radius2;
    vertex(sx, sy);
    sx = x + cos(a + halfAngle) * radius1;
    sy = y + sin(a + halfAngle) * radius1;
    vertex(sx, sy);
  }
  endShape(CLOSE);
}
```



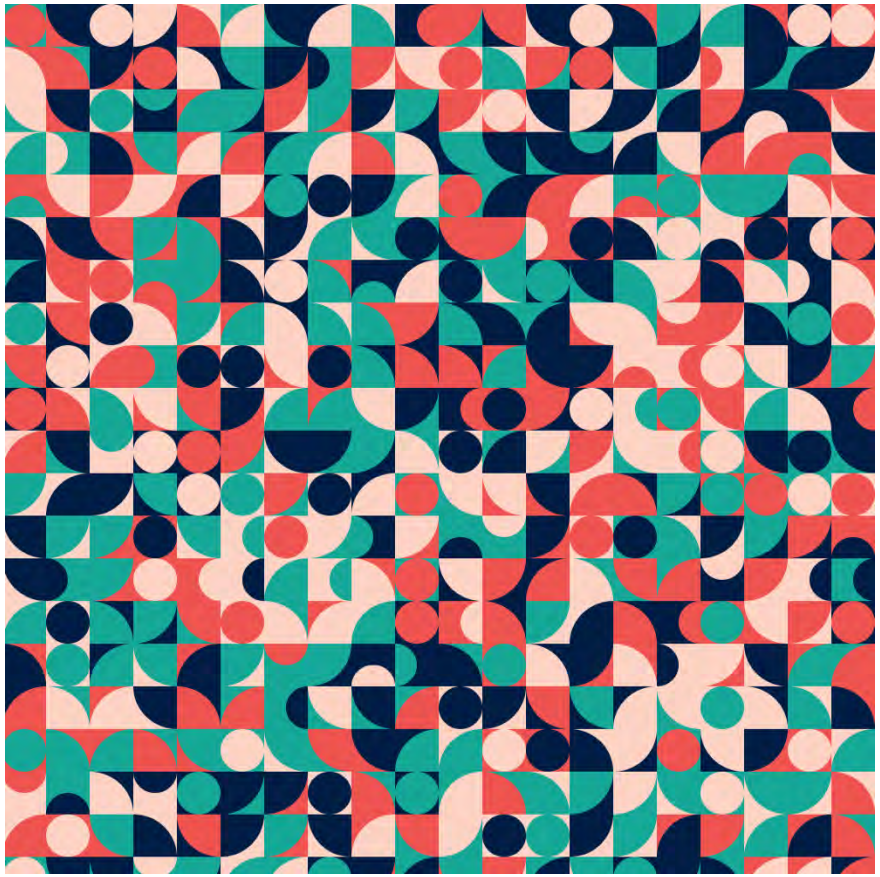
https://p5js.org/es/examples/form-star.html

User input and interaction



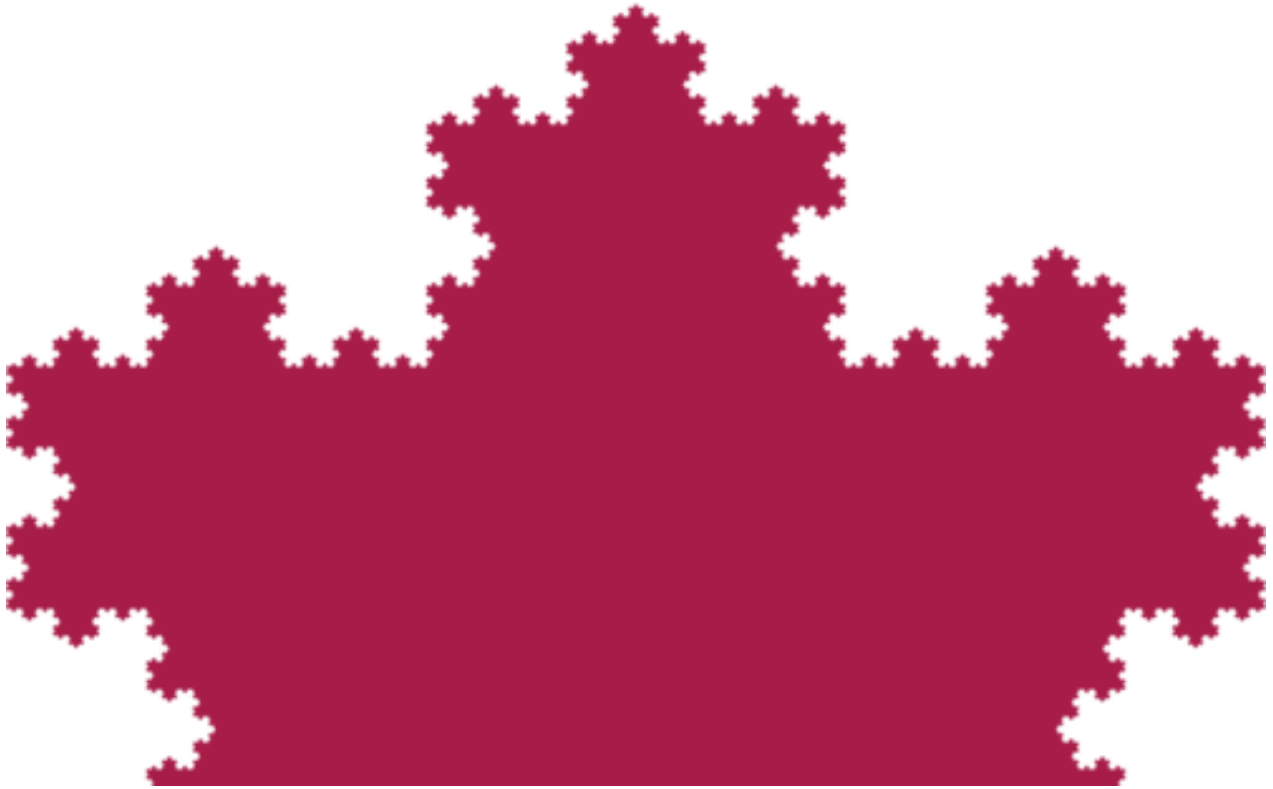
Possibilities with p5.js

- Geometric patterns



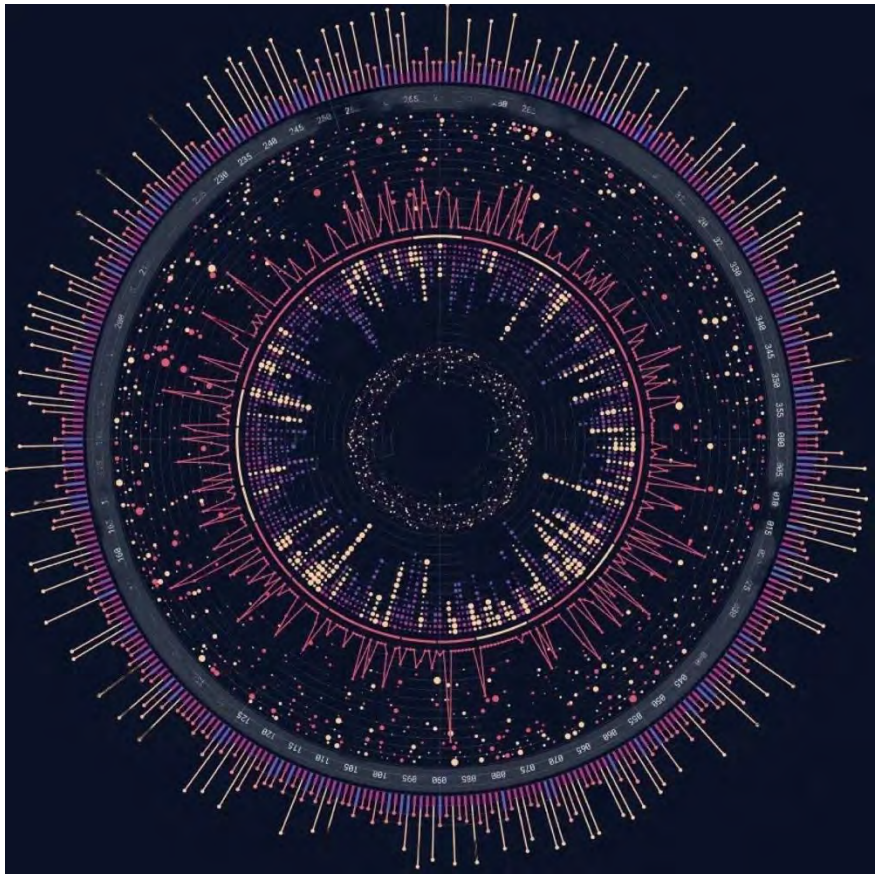
Possibilities with p5.js

- Fractals and iterated systems



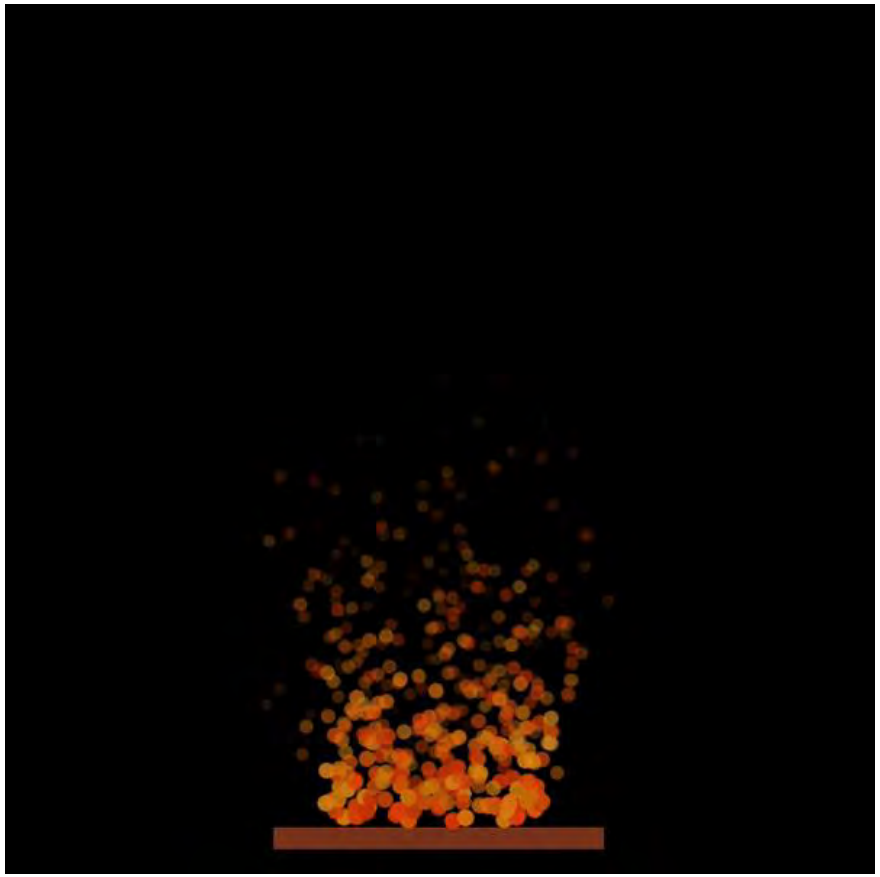
Possibilities with p5.js

- Visual representations of data



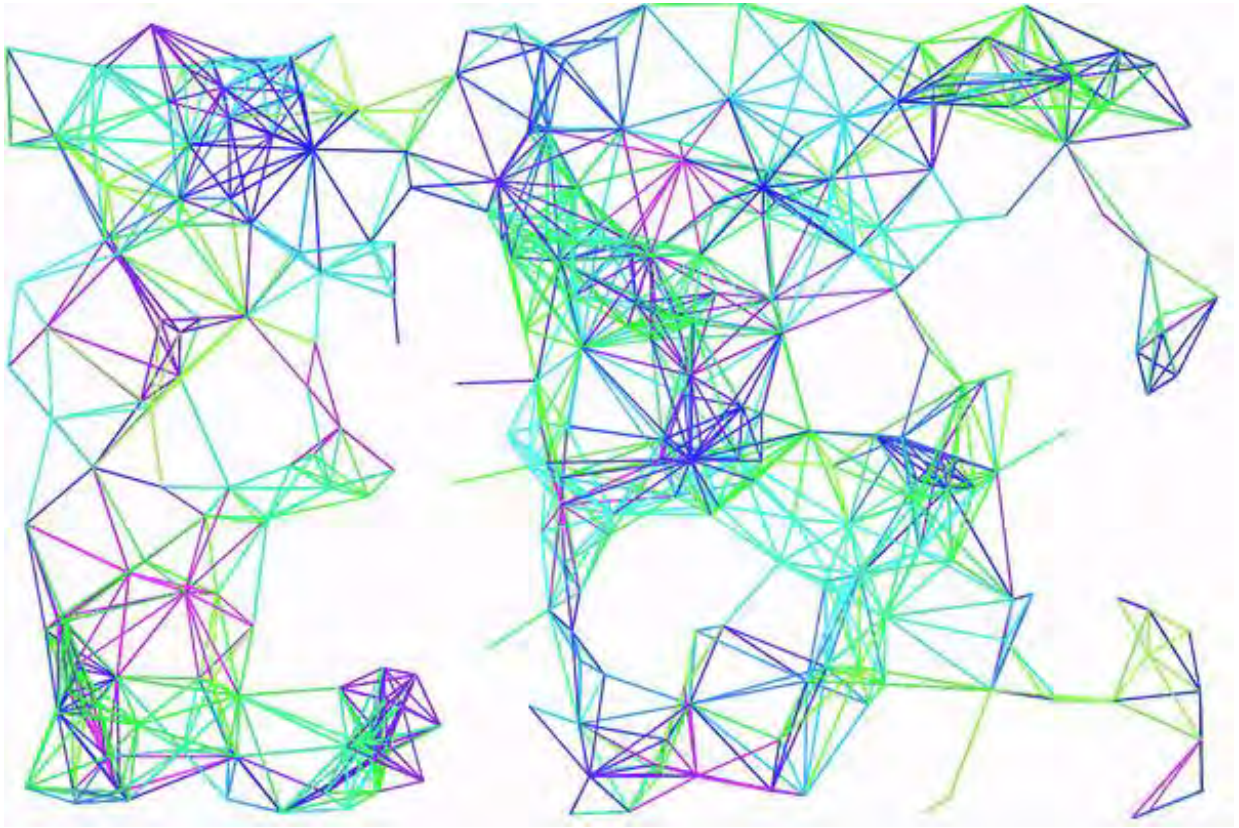
Possibilities with p5.js

- Simulations of nature



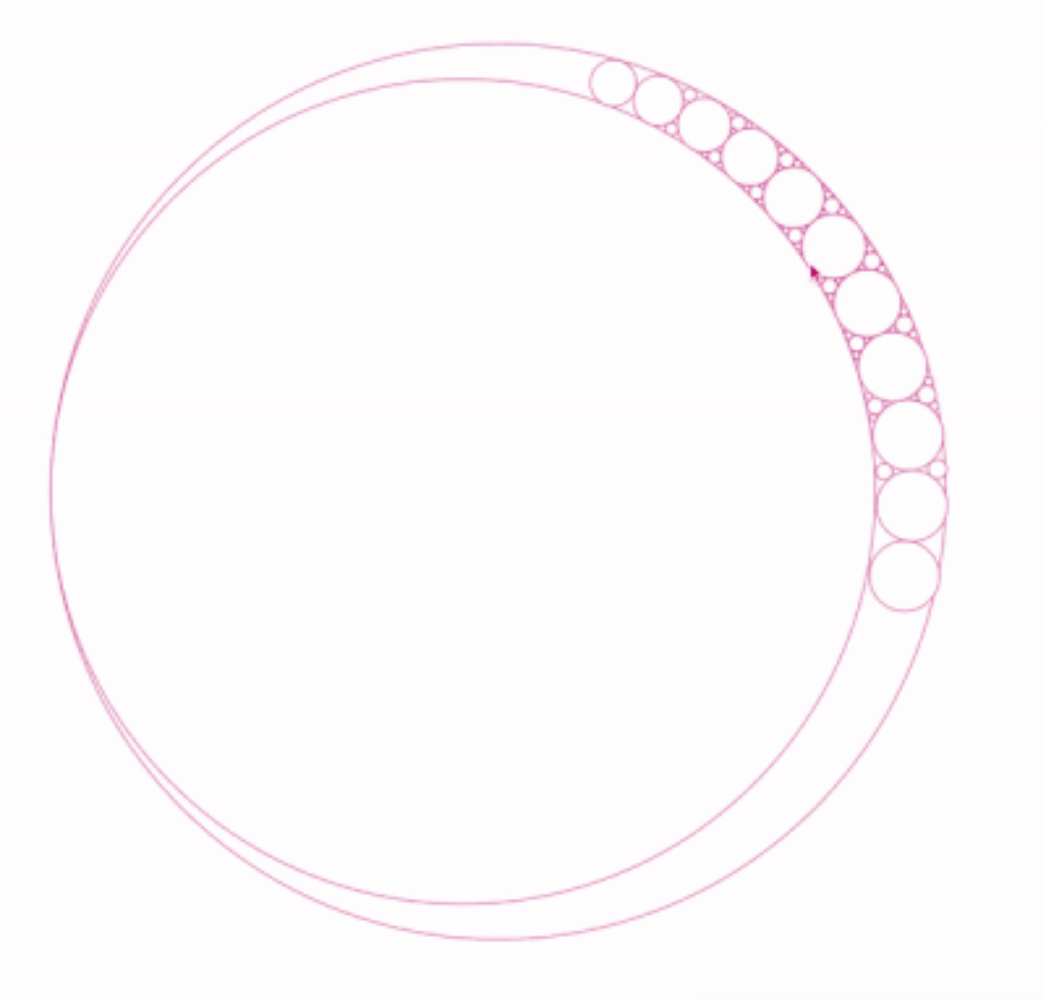
Possibilities with p5.js

- Randomness



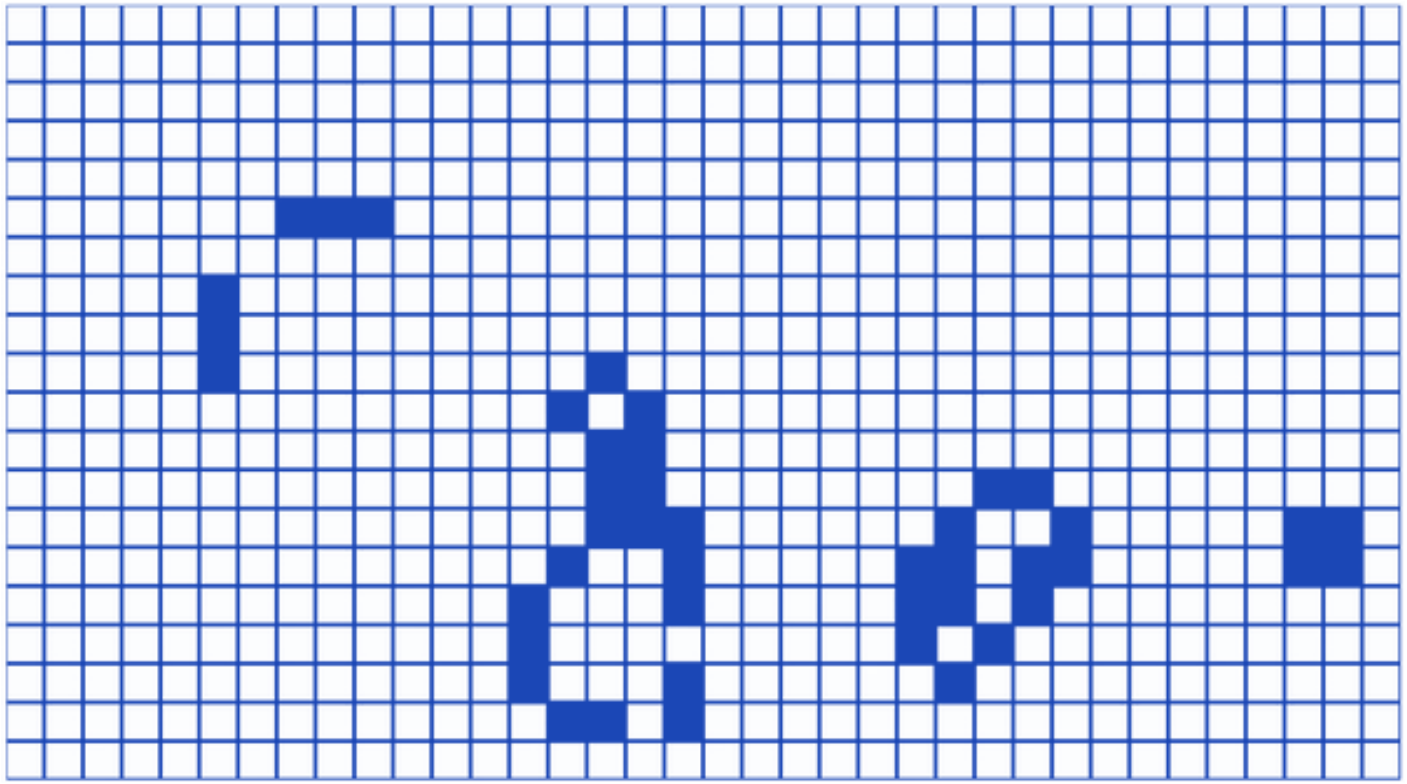
Possibilities with p5.js

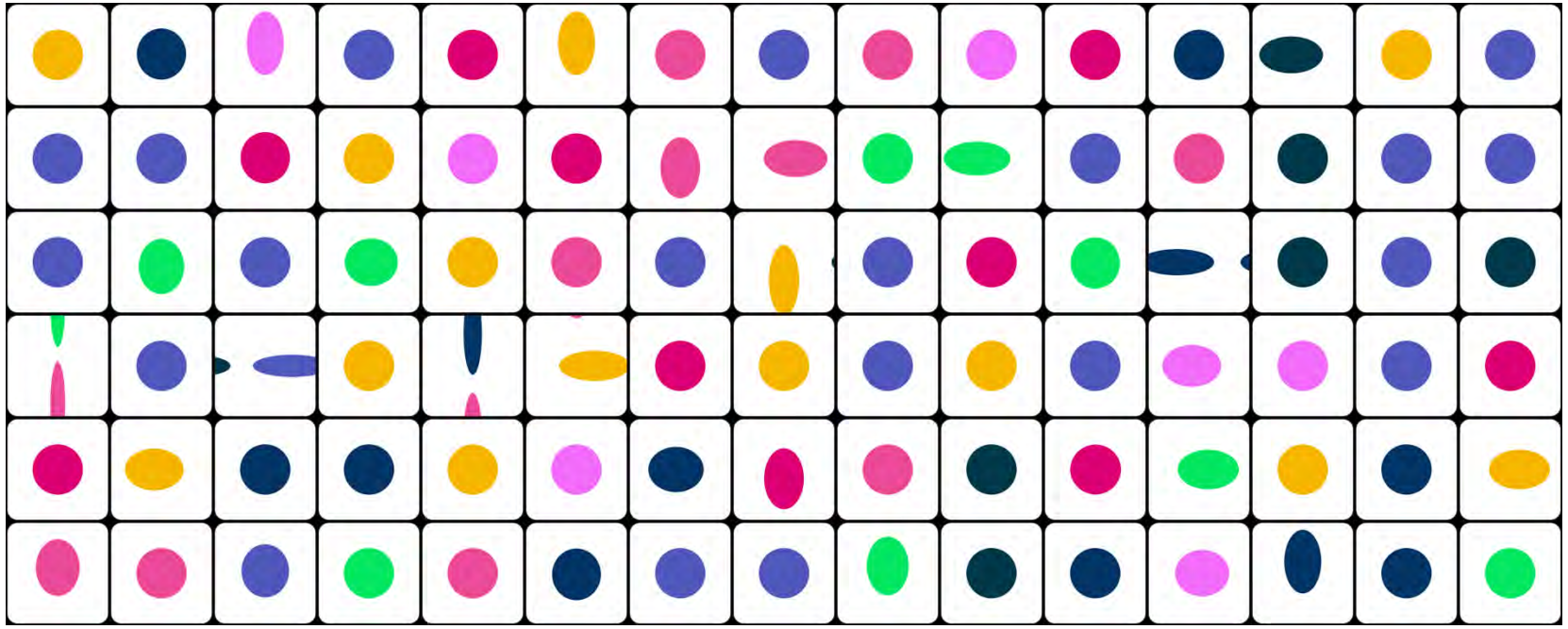
- Interactive generation



Possibilities with p5.js

- Cellular automats
  - Game of life

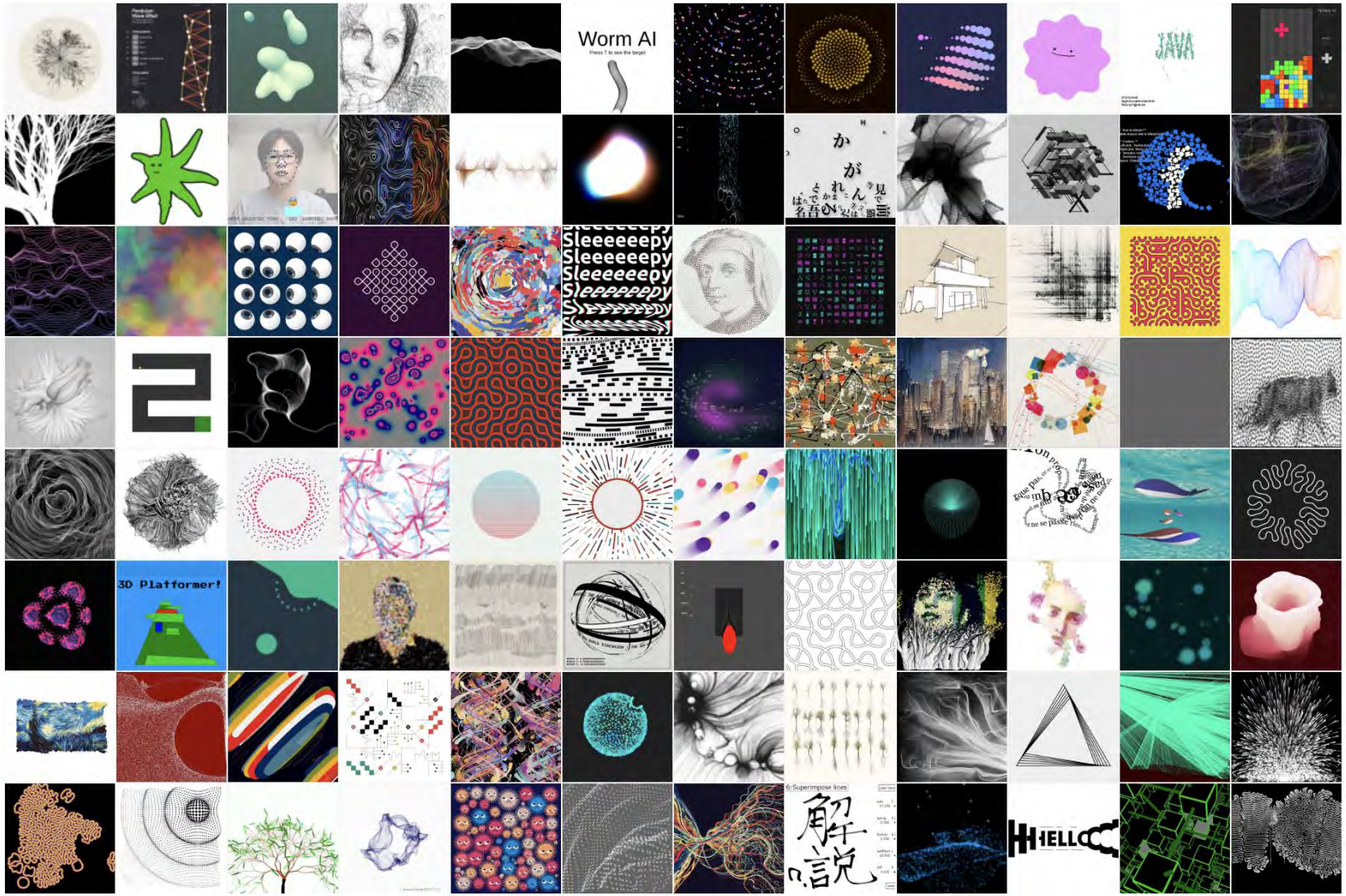




Experimentation Resources

Join thousands of creative programmers, follow their work and find inspiration for your next programming challenge.

frani.be

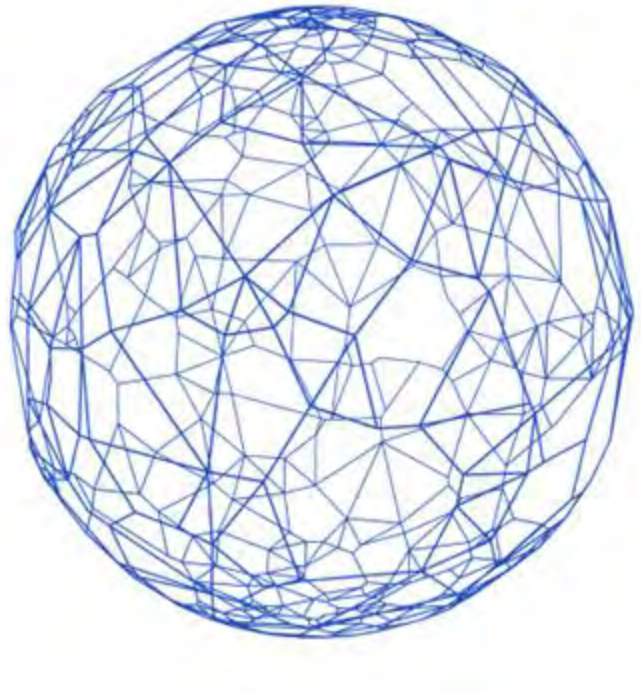


https://openprocessing.org/

Tangled

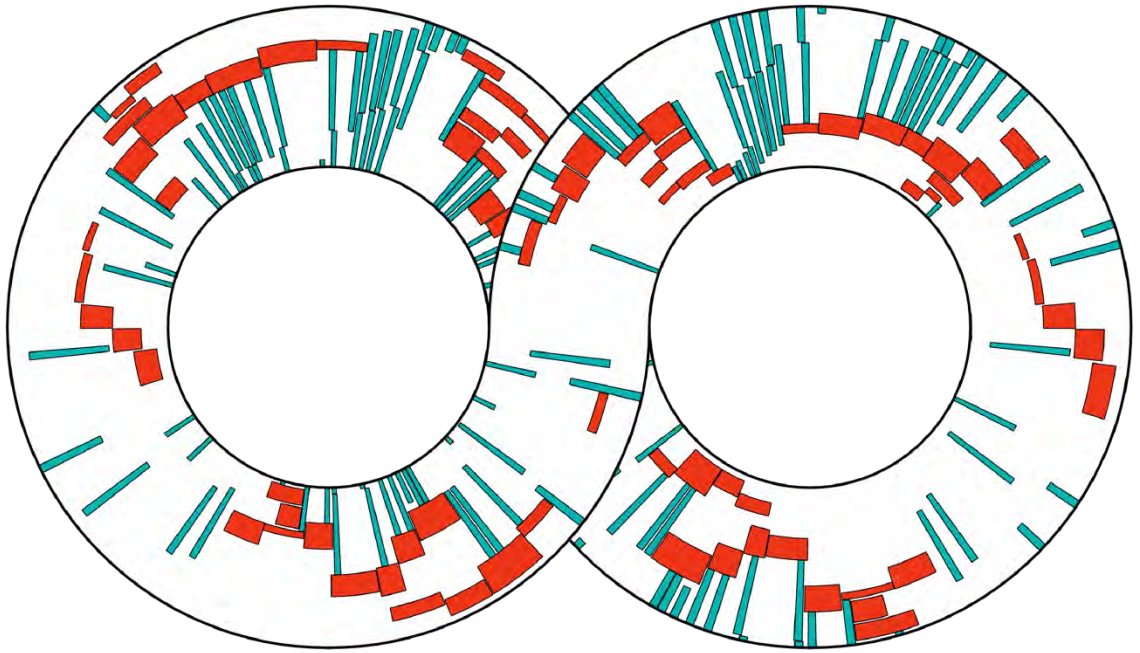
https://openprocessing.org/sketch/941007

Skeleton Sphere



https://openprocessing.org/sketch/2137668

Arcs and Arrows



https://openprocessing.org/sketch/2130848

🇨🇱

Conclusions

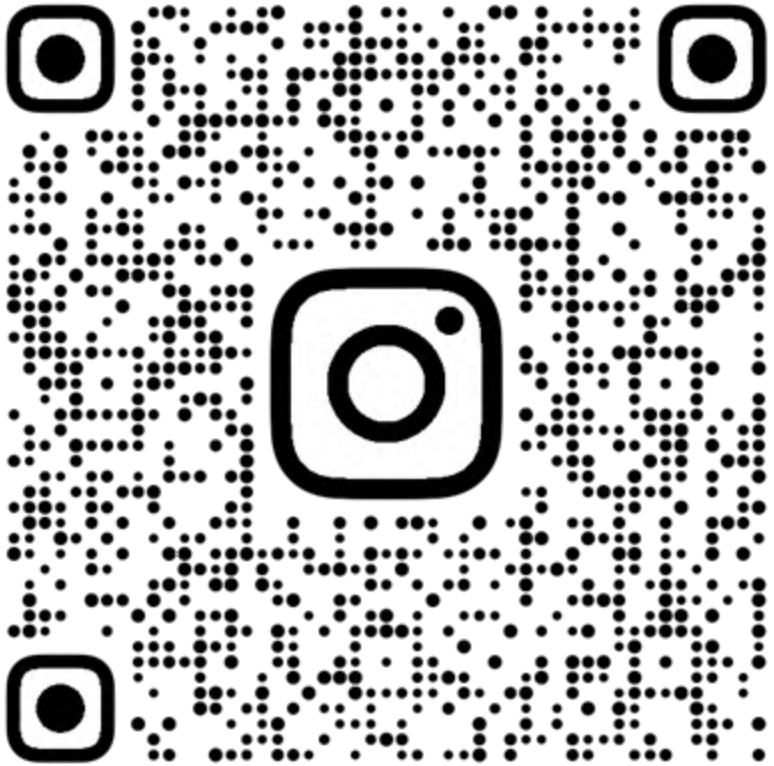
Did you know this kind of art? Did you know this library? After this talk… Are you motivated to be the new algorists of the decade?

frani.be

Bibliographical References

- https://www.verostko.com/algorithm.html
- https://www.hisour.com/es/algorithmic-art-12807/
- https://es.wikidat.com/info/arte-algoritmico
- https://docs.google.com/presentation/d/1ip8QnSZdF1PekqXNsBaXrsKUy65U5D767cgwd_nKli0/edit?usp=sharing
- https://openprocessing.org/
- https://p5js.org/es/
- https://www.freecodecamp.org/espanol/news/una-introduccion-al-arte-generativo-que-es-y-como-se-hace/
- https://editor.p5js.org/fede.santana/collections/UQkKGQbFD
- https://natureofcode.com/
- https://github.com/b2renger/Introduction_p5js

frani.be



@FRANI.BE