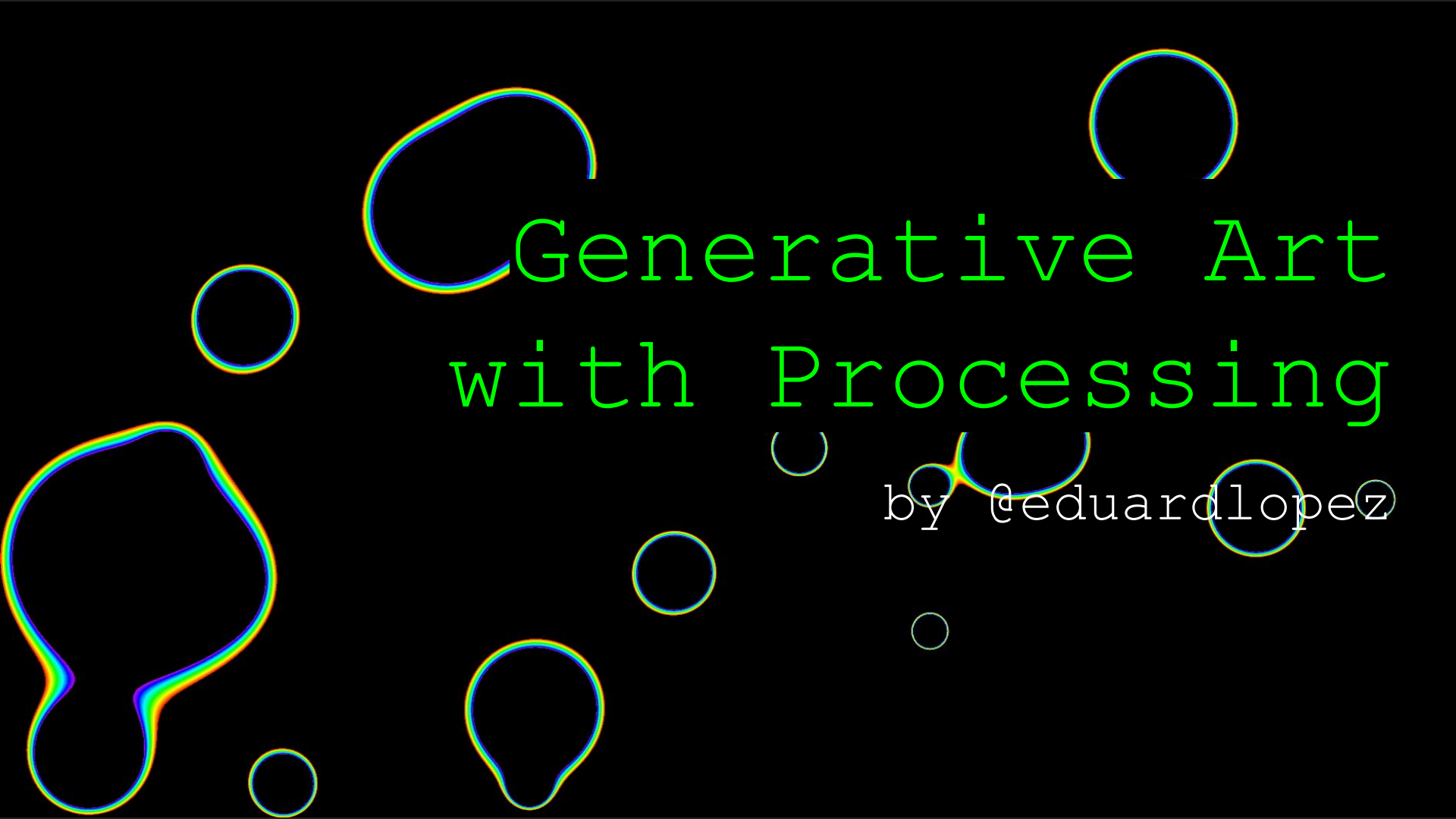
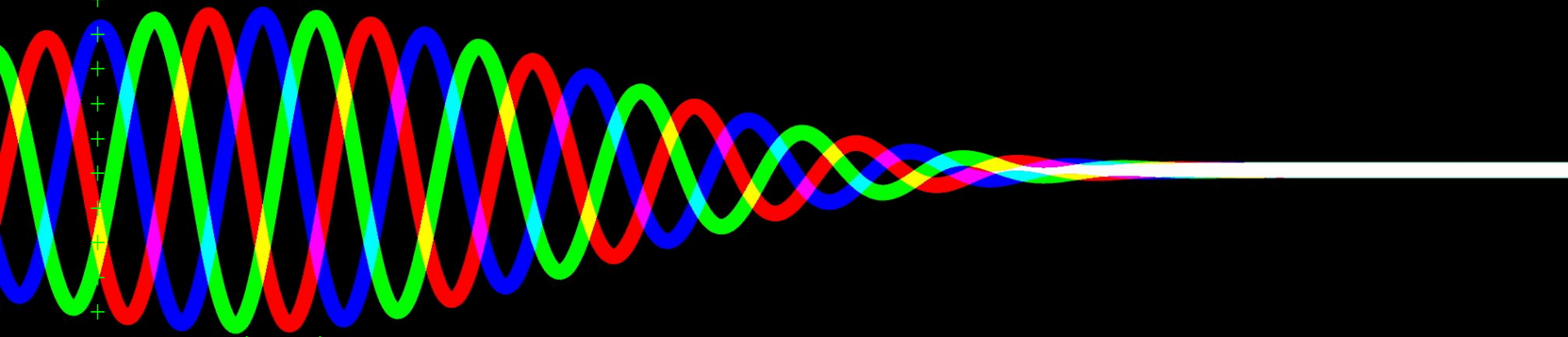

Generative Art
with Processing
by @eduardlopez

@eduardlopez

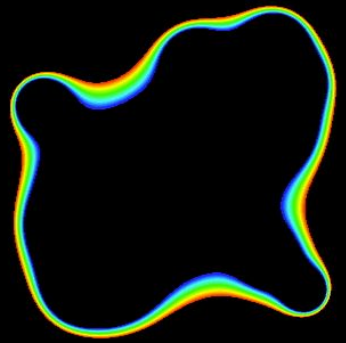
+ TecnoUAB Cofounder
+ 2017 UAB CS Graduate
+ 2017 Badi.com as Data and Machine Learning Engineer
+ 2021 Wallapop.com as Lead Data Engineer
+
+ linkedin.com/in/eduardlopezg
+ github.com/eduardlopez
+ twitter.com/eduardlopez
+ eduardlopez.com
+

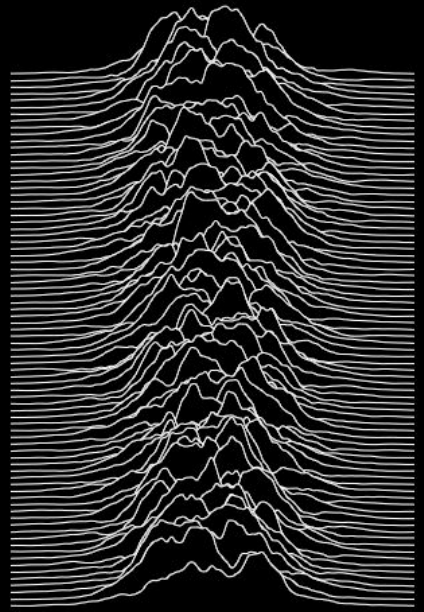
What’s Generative Art?

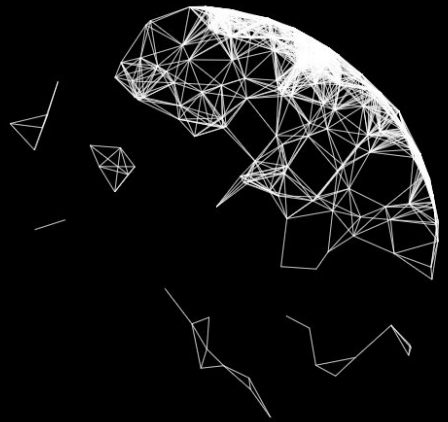
+ Digital assets*
+ Created with a computer*
+ You provide the rules -> boundaries
+ The computer follows them
+
+
+
+
+
+
+
+
+
+
+
+ algorithmic art
+ computer-generated art
+ synthetic media
+



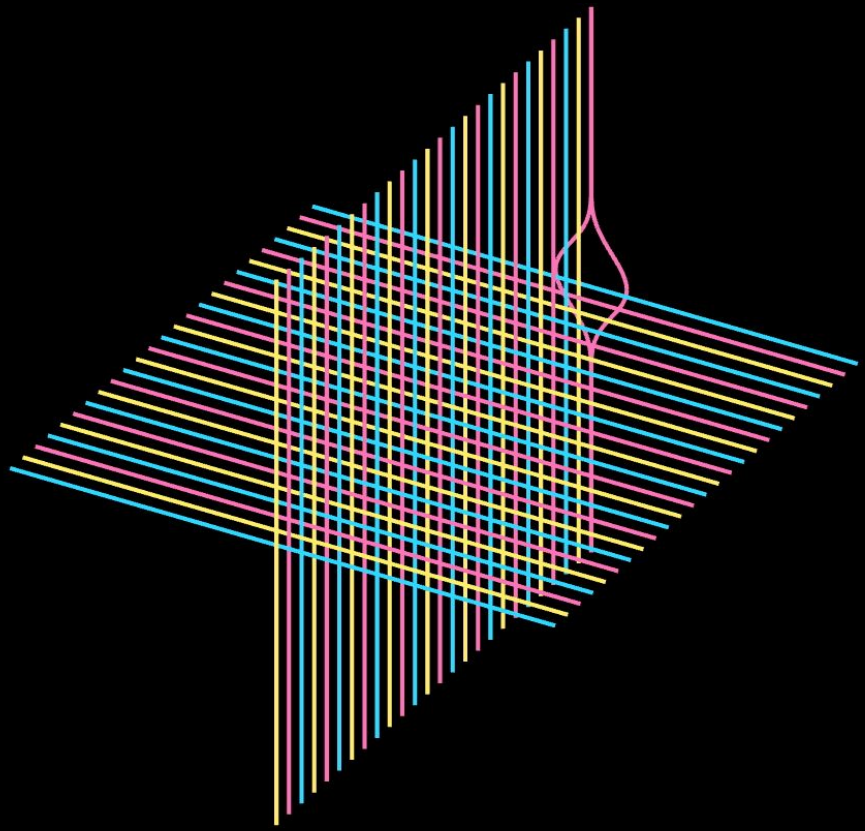
```
+      // Basile Pesin
+      // http://vertmo.github.io
+
+      // MetaBalls : p5.js implementation
+
+      var blobs = []
+
+      function setup() {
+        createCanvas(450, 450);
+                colorMode(HSB);
+        for (i = 0; i < 15; i++) blobs.push(new Blob(width/2,height/2));
+      }
+
+      function draw() {
+        background(0);
+        loadPixels();
+        for (x = 0; x < width; x++) {
+          for (y = 0; y < height; y++) {
+            let sum = 0;
+            for (i = 0; i < blobs.length; i++) {
+                let xdif = x - blobs[i].x;
+                                                let ydif = y - blobs[i].y;
+                let d = sqrt((xdif * xdif) + (ydif * ydif));
+              sum += 10 * blobs[i].r*10 / d;
+            }
+            if(sum > 230  && sum <255){
+                 set(x, y, color(map(sum,230,255,0,255), 255, 255));
+            }
+          }
+        }
+        updatePixels();
+
+        for (i = 0; i < blobs.length; i++) {
+          blobs[i].update();
+                            //blobs[i].show();
+        }
+      }
+
```



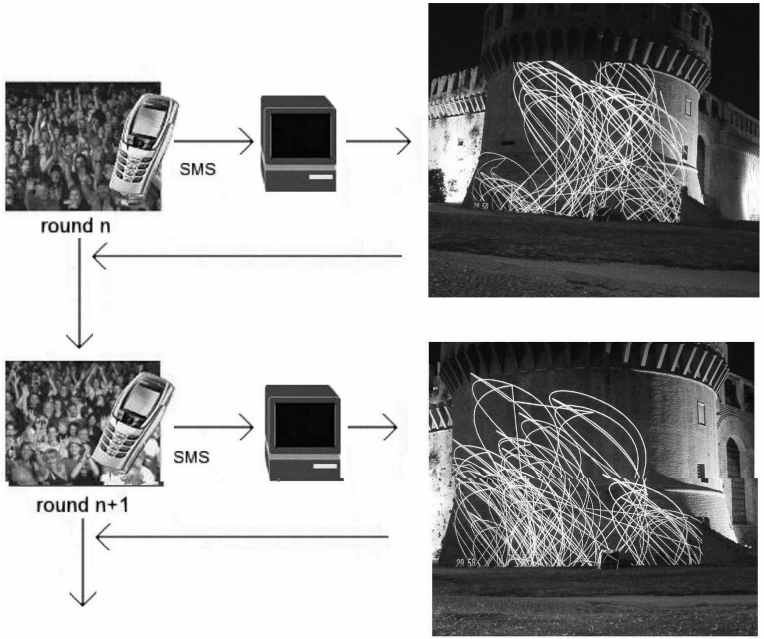


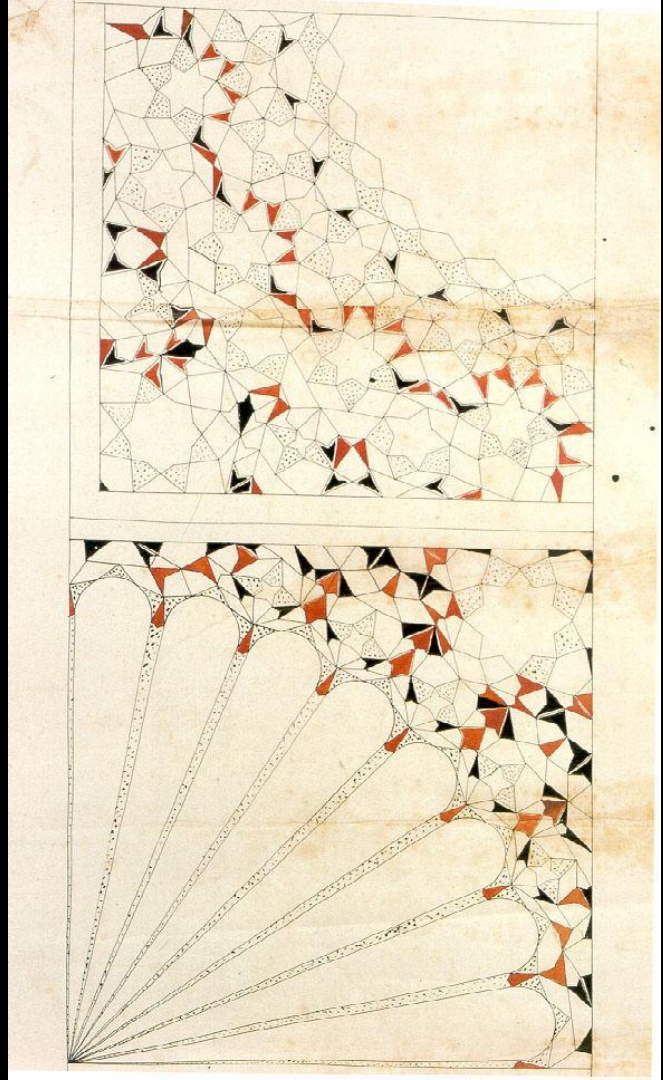


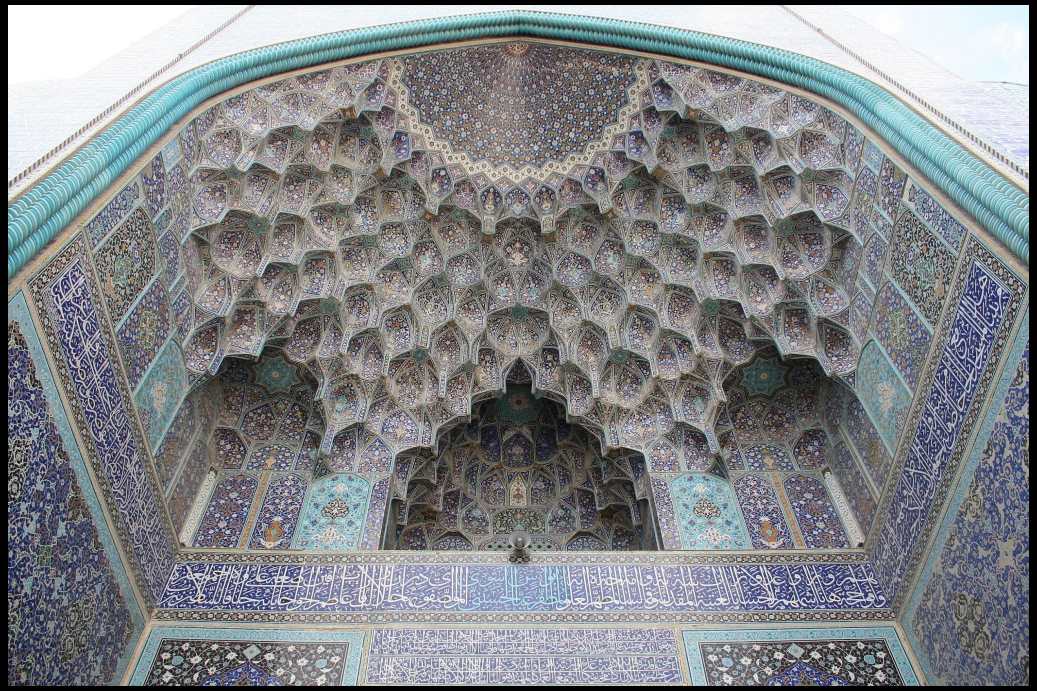
Sleeeeeeepy
Sleeeeeeepy
Sleeeeeeepy
Sleeeeeeepy
Sleeeeeeepy
Sleeeeeeepy
Sleeeeeeepy

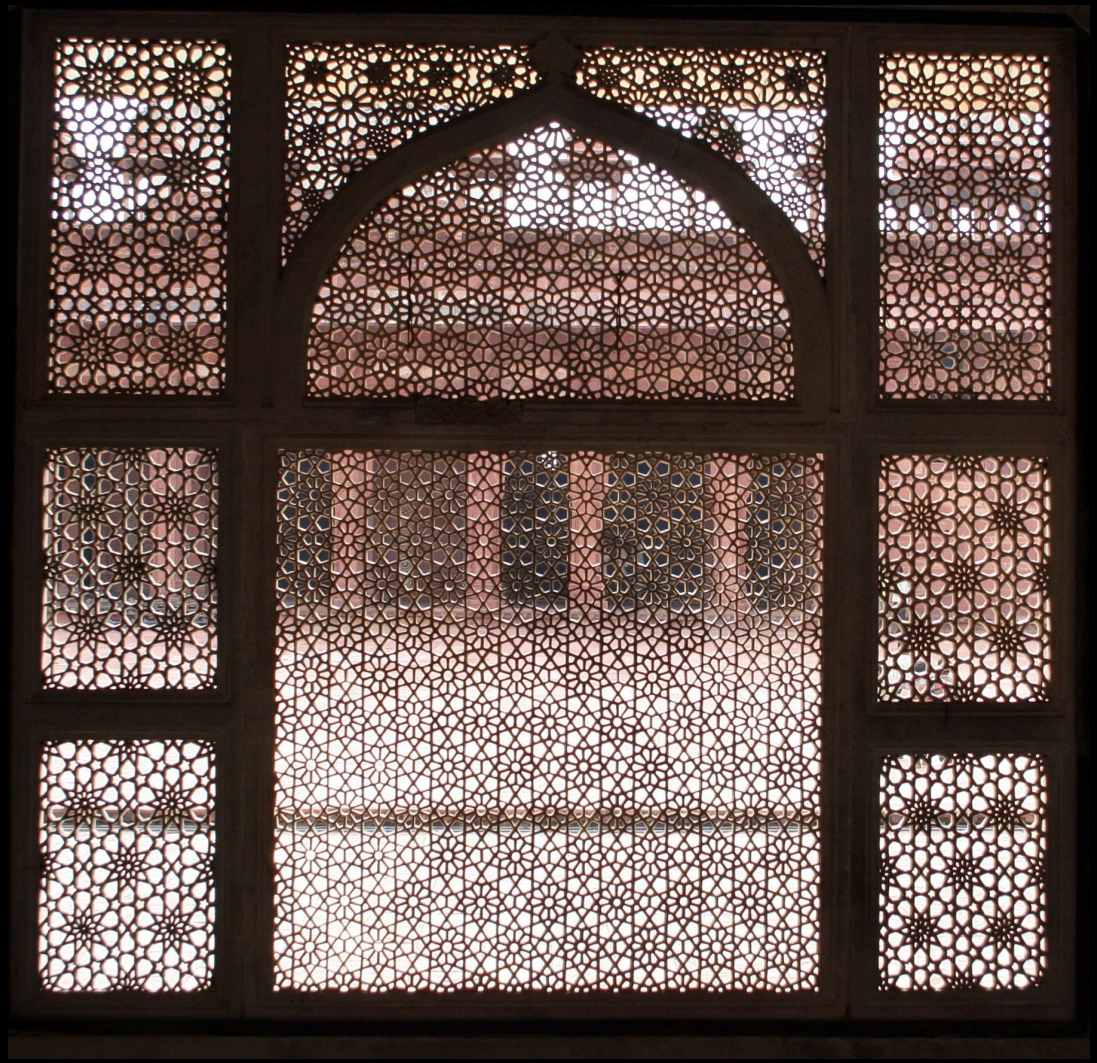


interactive + generative + public



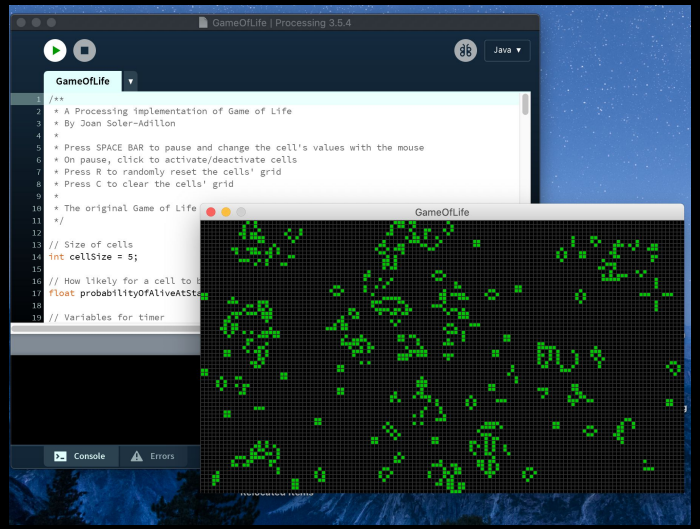






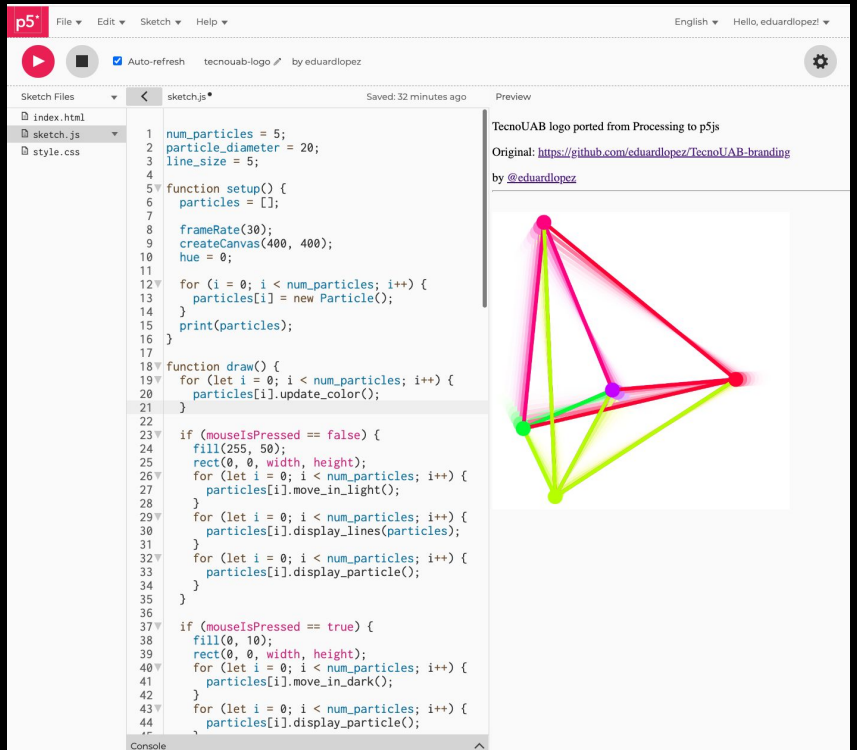
Tools | Processing.org

+ Graphical library + IDE
+ Based in Java with a layer of simplification
+ Self contained in an app, minimal friction to development
+ Main porting to javascript & python
+ Excelent learning materials: processing.org/tutorials/



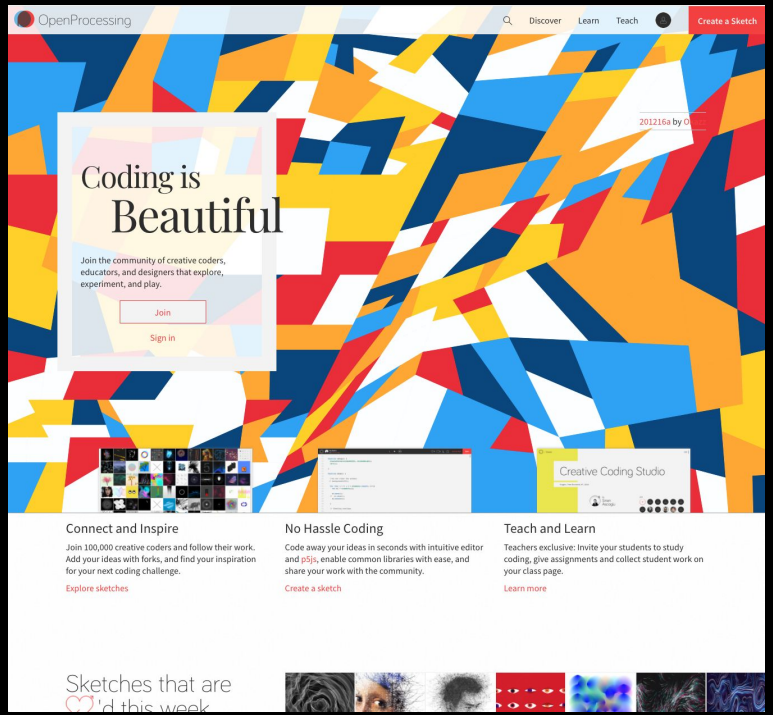
Tools | p5js.org

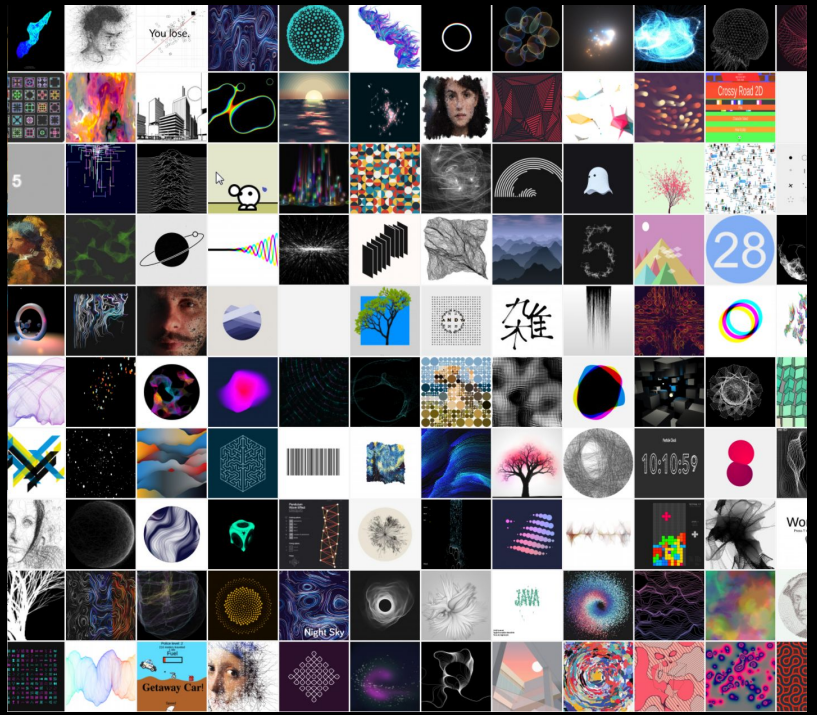
+ The best porting of Processing to javascript
+ Web browser editor: https://editor.p5js.org/
+ Learn: p5js.org/get-started



Resources | openprocessing.org

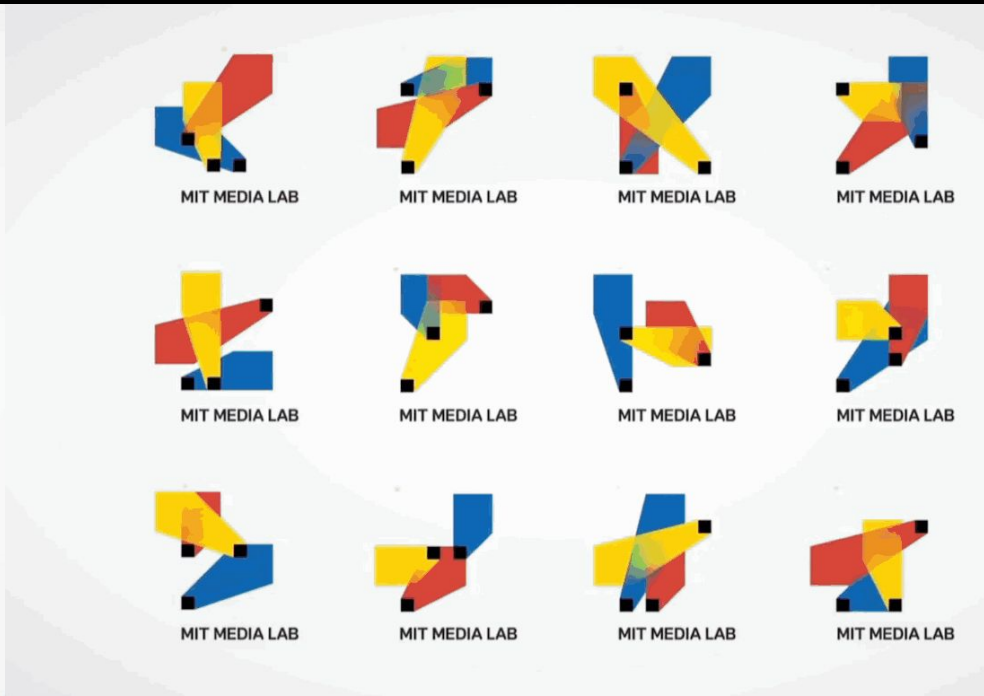
+ Inspiration, Sharing





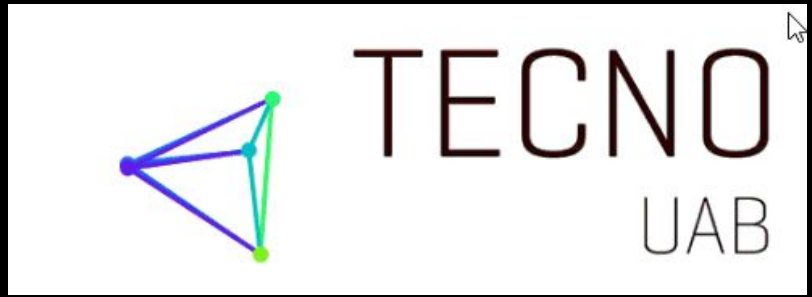
MIT Media Lab identity

+ 2011-2014



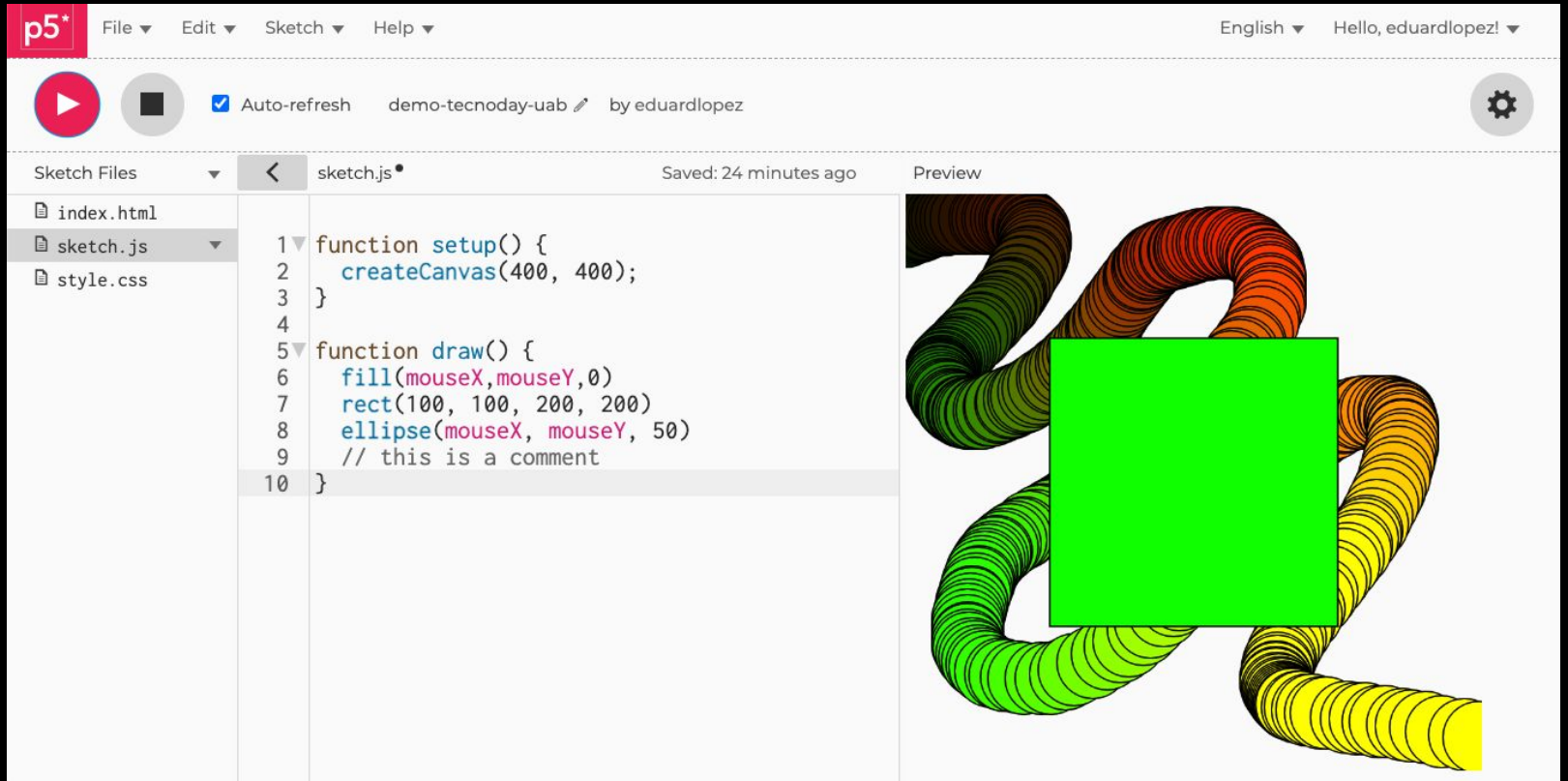
TecnoUAB Logo

- Programmed with Processing
  - https://github.com/eduardlopez/TecnoUAB-branding
- Ported to js with p5.js
  - https://editor.p5js.org/eduardlopez/sketches/euHm0shLE
- Original meaning intent
  - Particles: TecnoUAB members
  - Lines: Ideas interchange done thanks the association
  - Colors: Diversity of opinions
  - if mouse click
    - then dark -> no TecnoUAB -> no connections, no progress

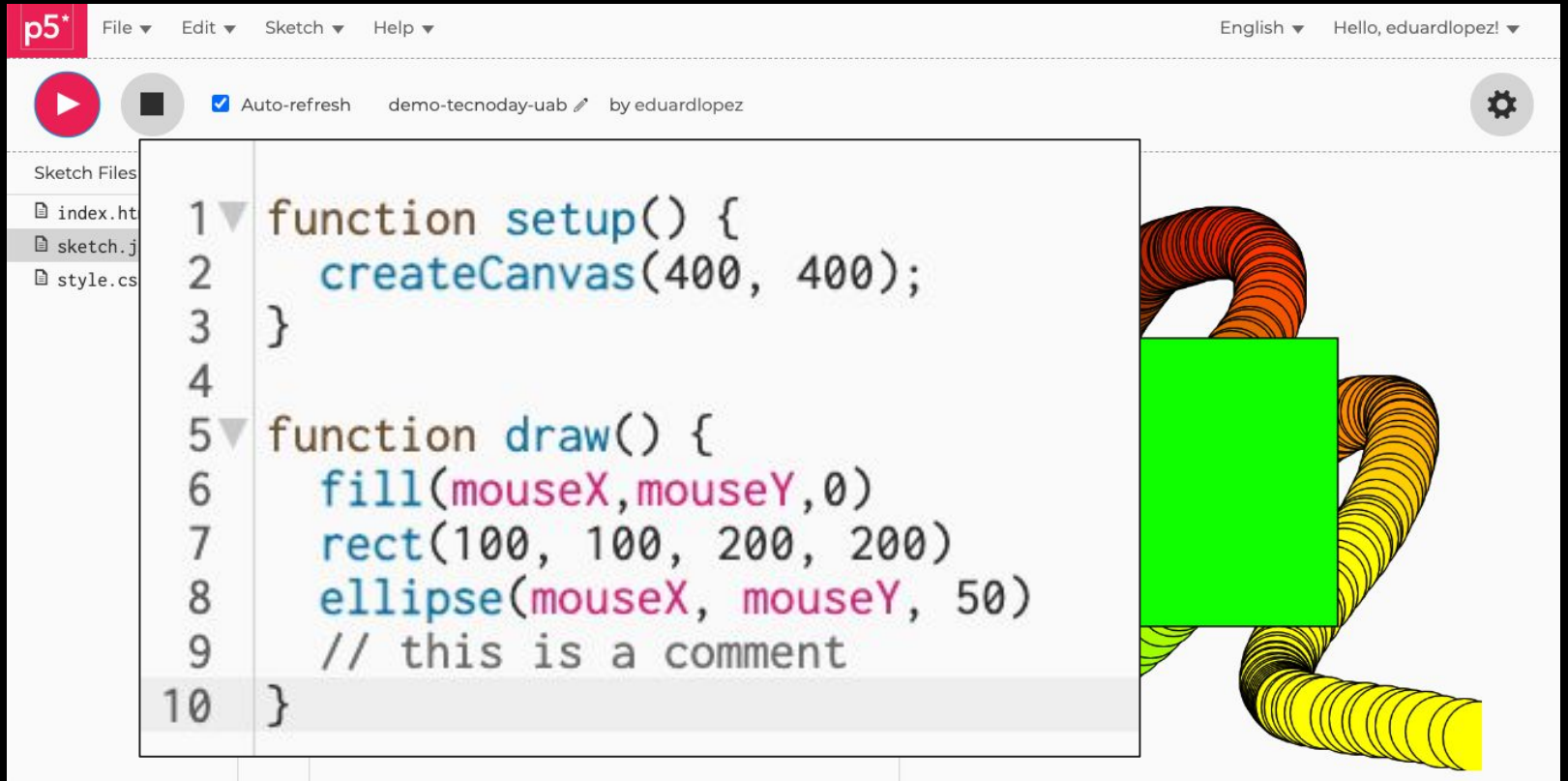


Let’s program!

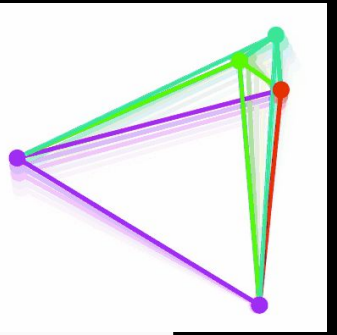
+ Go to https://editor.p5js.org/
+ Copy the lines
+ Comment and uncomment each line to see different results
+ Let's comment what's happening



+ Go to https://editor.p5js.org/
+ Copy the lines
+ Comment and uncomment each line to see different results
+ Let's comment what's happening



+ TecnoUAB p5.js logo
+ Interactive version:
https://editor.p5js.org/eduardlopez/sketches/euHm0shLE



```
num_particles = 5;
particle_diameter = 20;
line_size = 5;

function setup() {
  particles = [];

  frameRate(30);
  createCanvas(400, 400);
  hue = 0;

  for (i = 0; i < num_particles; i++) {
    particles[i] = new Particle();
  }
  print(particles);
}

function draw() {
  for (let i = 0; i < num_particles; i++) {
    particles[i].update_color();
  }

  if (mouseIsPressed == false) {
    fill(255, 50);
    rect(0, 0, width, height);
    for (let i = 0; i < num_particles; i++) {
      particles[i].move_in_light();
    }
    for (let i = 0; i < num_particles; i++) {
      particles[i].display_lines(particles);
    }
    for (let i = 0; i < num_particles; i++) {
      particles[i].display_particle();
    }
  }

  if (mouseIsPressed == true) {
    fill(0, 10);
    rect(0, 0, width, height);
    for (let i = 0; i < num_particles; i++) {
      particles[i].move_in_dark();
    }
    for (let i = 0; i < num_particles; i++) {
      particles[i].display_particle();
    }
  }
}

function mouseReleased() {
  for (let i = 0; i < num_particles; i++) {
    particles[i].init_dirs();
  }
}

class Particle {
  constructor() {
    this.diameter = particle_diameter;
    this.line_size = line_size;

    this.min_width = this.diameter;
    this.max_width = width - this.diameter;

    this.min_height = this.diameter;
    this.max_height = height - this.diameter;

    this.x = random(this.min_width, this.max_width
    this.y = random(this.min_height, this.max_heig
    this.dir_x = 0;
    this.dir_y = 0;
    this.init_dirs();
    this.speed = 1;
    this.hue = random(0, 100);
    this.color = color(this.hue, 100, 100);
  }

  init_dirs() {
    this.dir_x = random(-10, 10);
    this.dir_y = random(-10, 10);
  }

  update_color() {
    if (this.hue > 100) {
      this.hue = 0;
    } else {
      this.hue += 1;
    }
    colorMode(HSB, 100);
    this.color = color(this.hue, 100, 100);
  }

  move_in_light() {
    if (this.x <= this.min_width) {
      this.x = this.min_width + 1;
      this.dir_x = this.dir_x * -1;
    }
    if (this.x >= this.max_width) {
      this.x = this.max_width - 1;
      this.dir_x = this.dir_x * -1;
    }

    if (this.y <= this.min_height) {
      this.y = this.min_height + 1;
      this.dir_y = this.dir_y * -1;
    }
    if (this.y >= this.max_height) {
      this.y = this.max_height - 1;
      this.dir_y = this.dir_y * -1;
    }

    this.x = this.x + this.dir_x;
    this.y = this.y + this.dir_y;
  }

  move_in_dark() {
    this.dir_x += random(2.0) - 1.0;
    this.dir_x *= 0.96;
    this.dir_y += random(2.0) - 1.0;
    this.dir_y *= 0.96;
    this.move_in_light();
  }

  display_lines(particles) {
    strokeWeight(this.line_size);
    stroke(this.color);
    for (let i = 0; i < num_particles; i++) {
      line(this.x, this.y, particles[i].x, particles[i].y);
    }
  }

  display_particle() {
    noStroke();
    fill(this.hue, 100, 100);
    ellipse(this.x, this.y, this.diameter, this.diameter);
  }
}
```

Sources

https://openprocessing.org/sketch/838276
https://vimeo.com/20488585
https://openprocessing.org/sketch/683686
https://openprocessing.org/sketch/434620
https://openprocessing.org/sketch/917861
https://openprocessing.org/sketch/733725
https://openprocessing.org/sketch/940954
https://openprocessing.org/sketch/941636
https://openprocessing.org/sketch/402961
https://aiartists.org/generative-art-design
https://www.youtube.com/watch?v=DZphIScEMtY
https://processing.org/
https://openprocessing.org/
https://en.wikipedia.org/wiki/Processing_(programming_language)
https://en.wikipedia.org/wiki/Generative_art
https://proyectoidis.org/maurizio-bolognini/