RUSTFEST
GLOBAL 2021
Rust in Arts Edition

What is this thing called Generative Art?

A (really) quick interdimensional flight on a Rust-y spaceship!

Francesco Cauteruccio, Ph.D.
f.cauteruccio@gmail.com

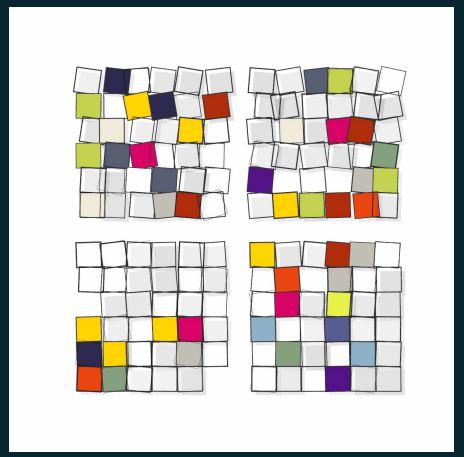
println!("Who am I?");

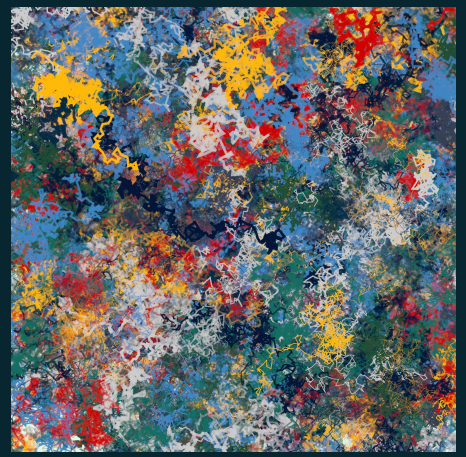
Francesco Cauteruccio

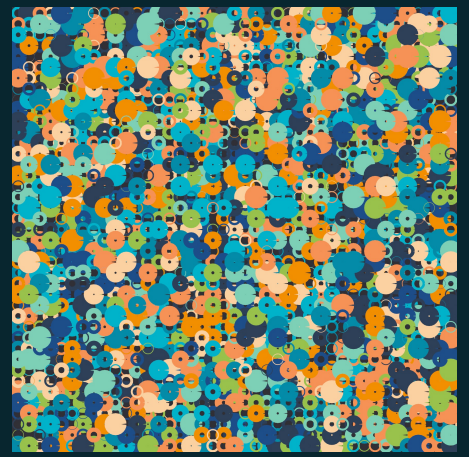
- Ph.D. in Computer Science
- Research fellow @ University of Calabria (Italy)

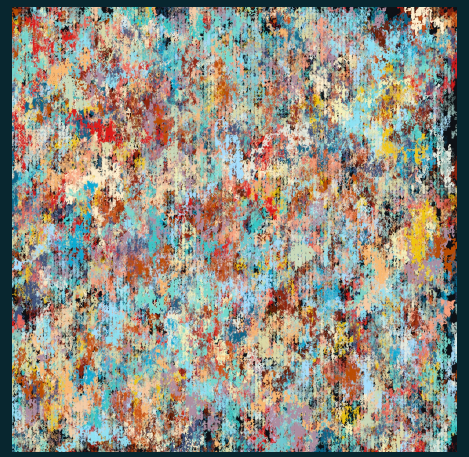
`instagram.com/imnotprovable`

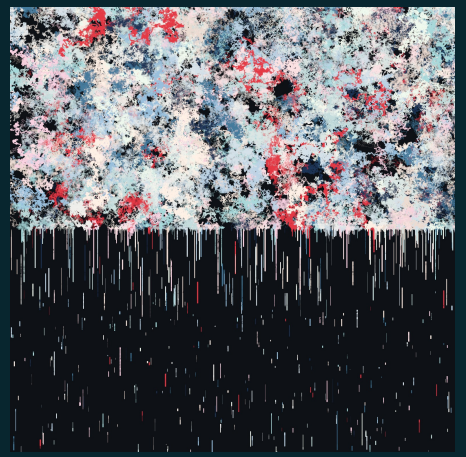
- Showcase of generative art
- Mostly made in Rust
- `github.com/finalfire/imnotprovable`











println!("About this talk");

- Quick overview of generative art
- Meet **nannou**, a framework for generative art in Rust
- Let’s make art!

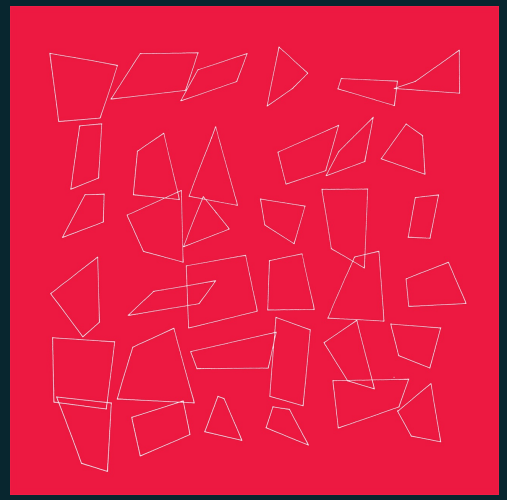
println!("What is Generative Art?");

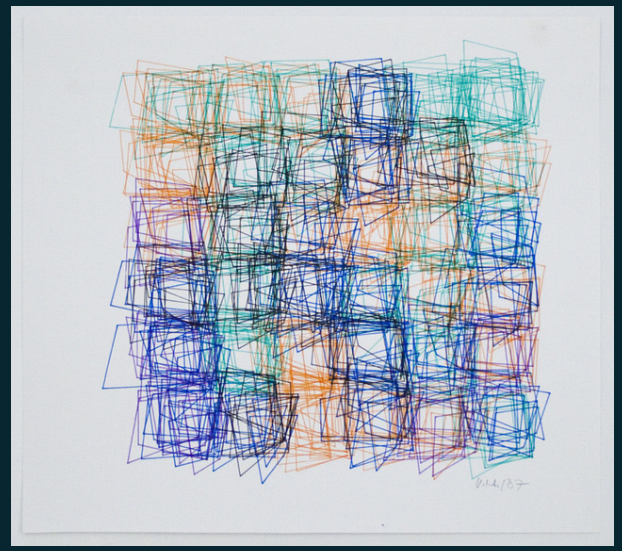
> Generative art refers to art that in whole or in part has been created with the use of an *autonomous* system.
>
> *Generative Art (wikipedia.org)*

("synonyms" are *creative coding*, *creative programming*, *algorithmic art*, …)

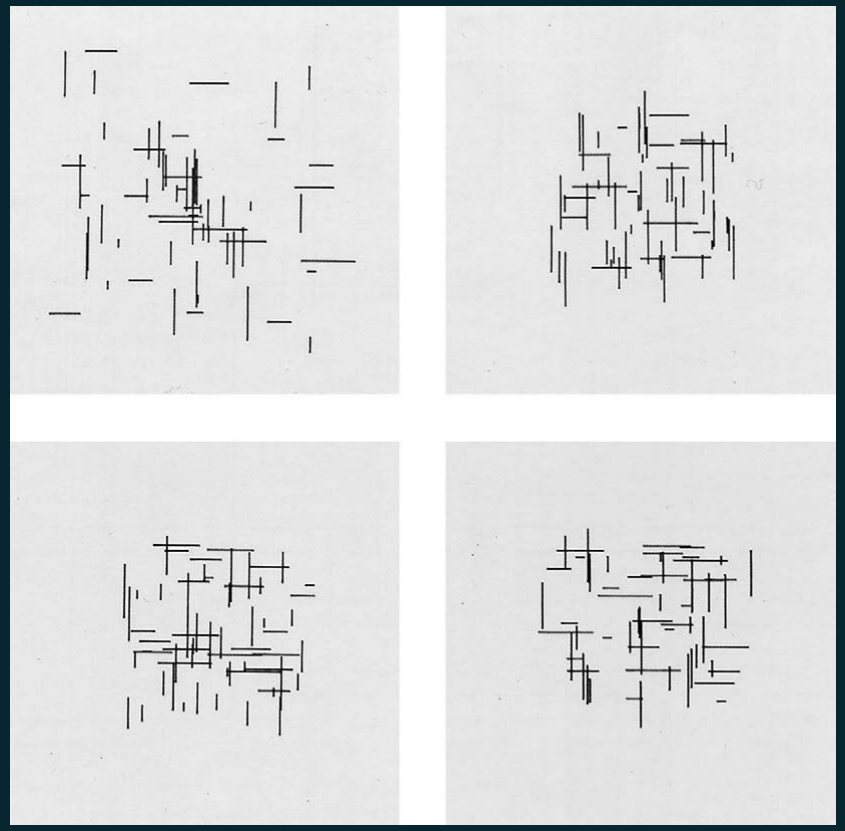
println!("Overview of Generative Art");

- Authors
- Learning
- Community
- Frameworks





Vera Molnár, Georg Nees, Frieder Nake, Brian Eno, …



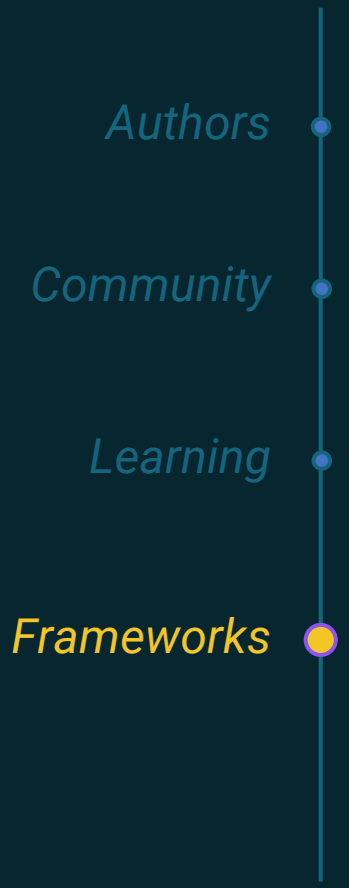
println!("Overview of Generative Art");

- Framework documentation
  - Generative Art is a nice way to learn a new language
- Youtube
  - Take a look at The Coding Train by Daniel Shiffman
- Awesome Creative Coding on Github
  - `github.com/terkelg/awesome-creative-coding`
- Blog posts

println!("Overview of Generative Art");

- Communities
  - Reddit `/r/generativeart`
  - Instagram and Twitter `#generativeart`
- `openprocessing.org`
  - Get inspired and try to recreate similar work in your framework of choice

println!("Overview of Generative Art");



- Several frameworks in several languages:
  - Processing (Java)
  - p5js (Javascript)
  - openframeworks (C++)
  - OPENRNDR (Kotlin)
  - Quill (Clojure)
  - **nannou (Rust)**

println!("nannou");

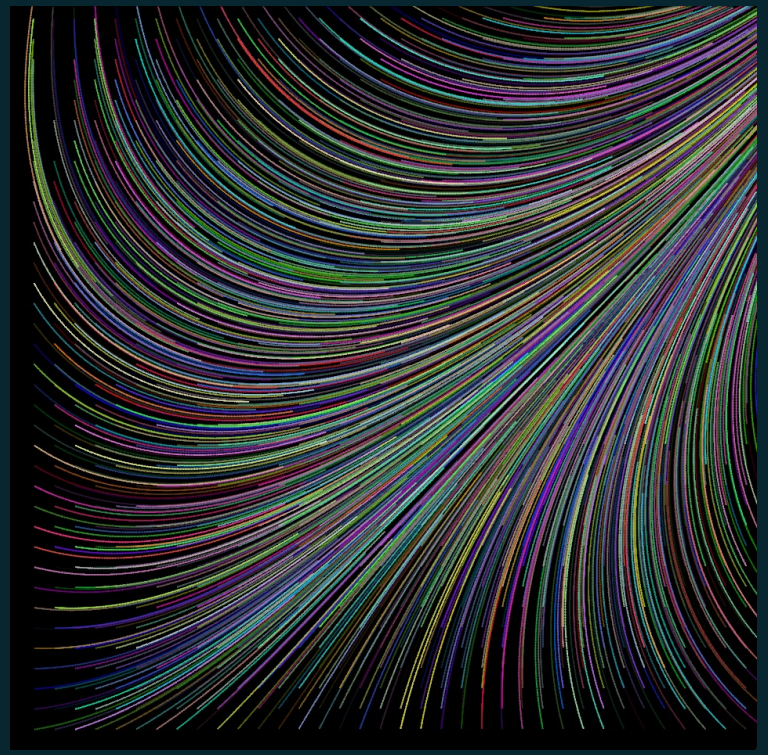
nannou https://nannou.cc/

- open source creative-coding framework for Rust
- full suite (graphics, audio, I/O, protocols, …)
- takes full advantage of Rust features
- follows a model-view-controller (MVC) pattern
- `nannou::sketch()` and `nannou::app()`
  - sketch is the fastest way to create a visualization
  - app is a more general way to create a visualization
  - grab the template from `bit.ly/nannou-skeletons`

lets_do_something()

Let’s recreate a simple *flow field*

- The goal is to make our hands dirty with nannou
- There are several ways to make such piece
  - *Do whatever you want, we are doing art!*

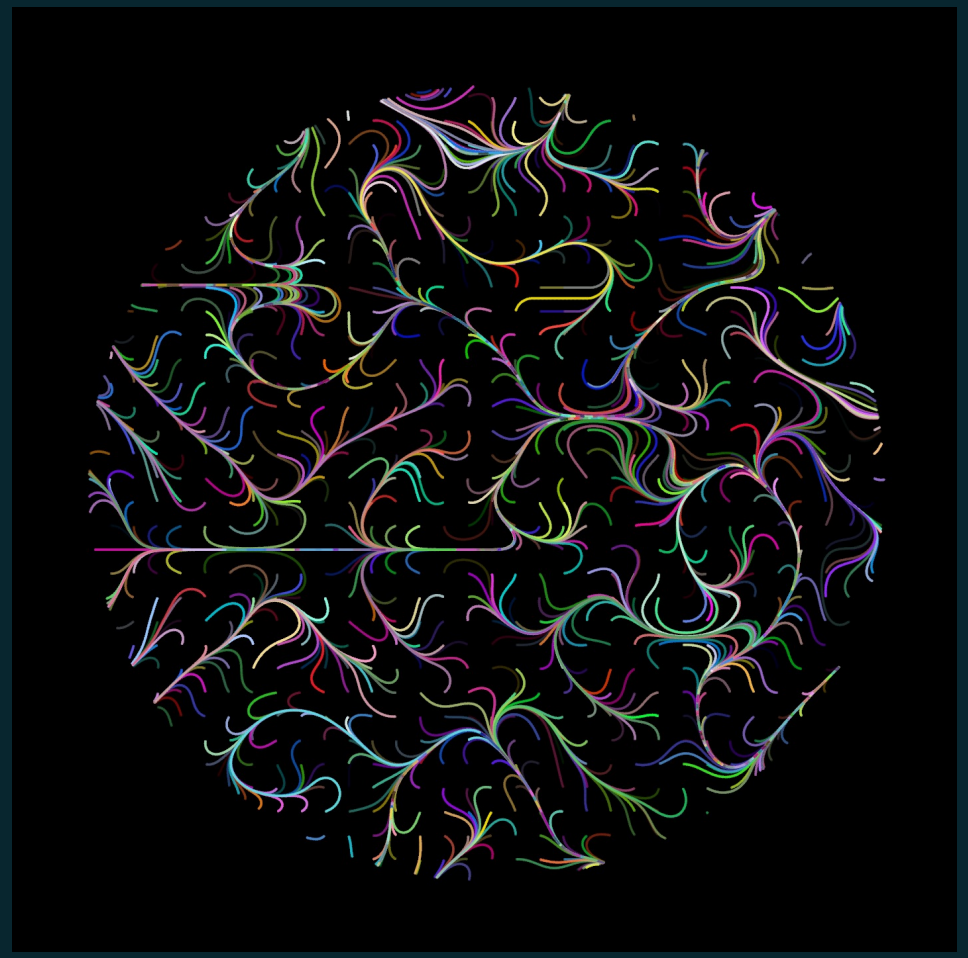


Three simple steps

1. Lay a grid of points
2. Give each point a direction
3. Make them move!

variation!("Circle mask");

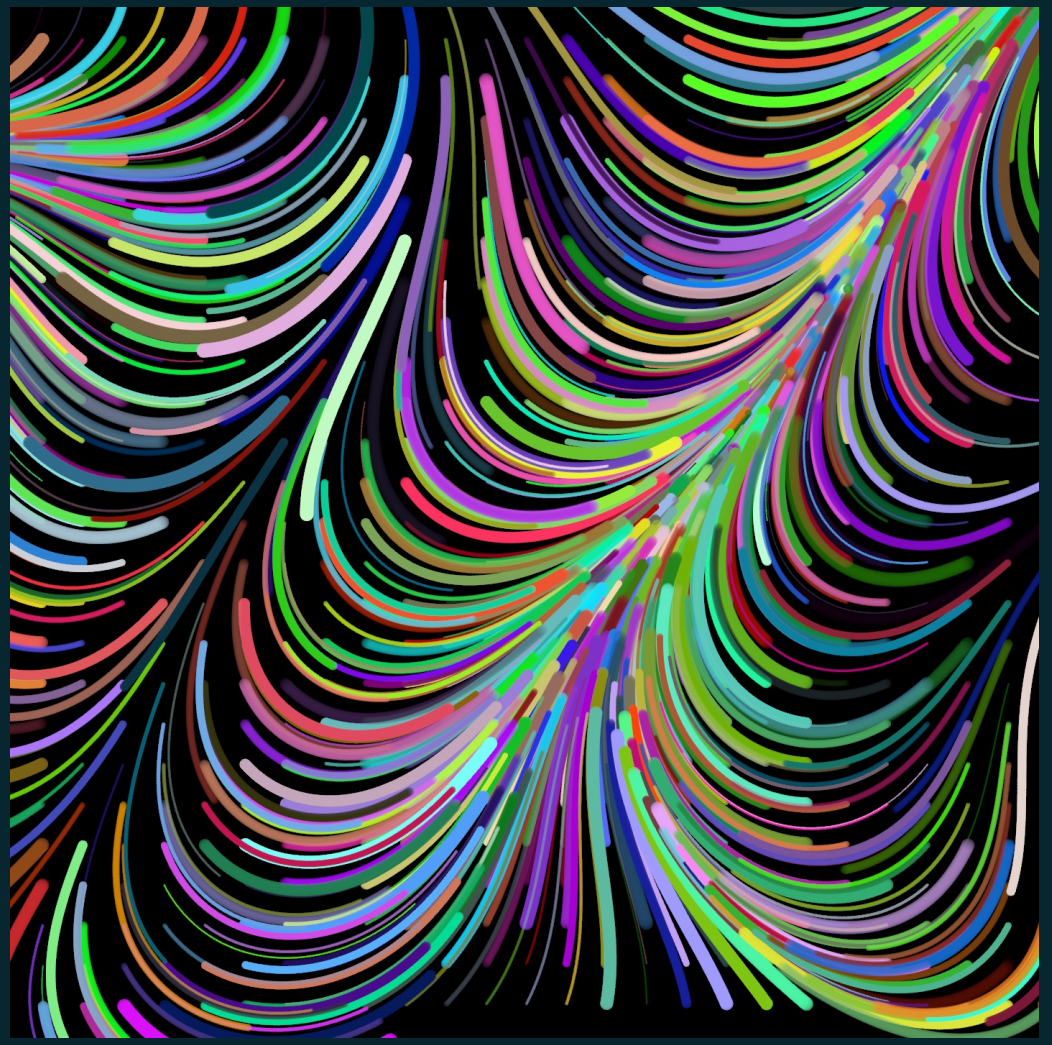
```
let origin = pt2(0.0, 0.0);
for point in &model.points {
    if point.coordinates.distance(origin) < 300.0 {
        // draw the point...
    }
}
```



variation!("Size of the points");

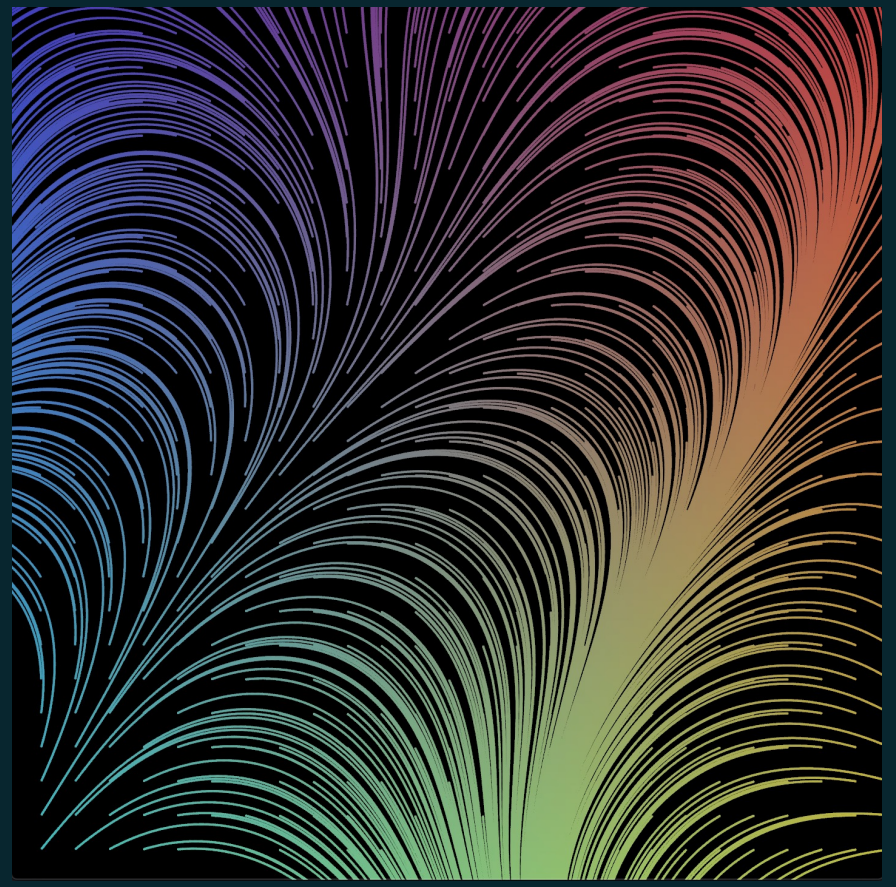
```
struct Point {
    coordinates: Point2,
    size: f32,
    color: Rgba
}
```

```
let size = random_range(1.0, 10.0);
points.push(Point { coordinates, size, color });
```



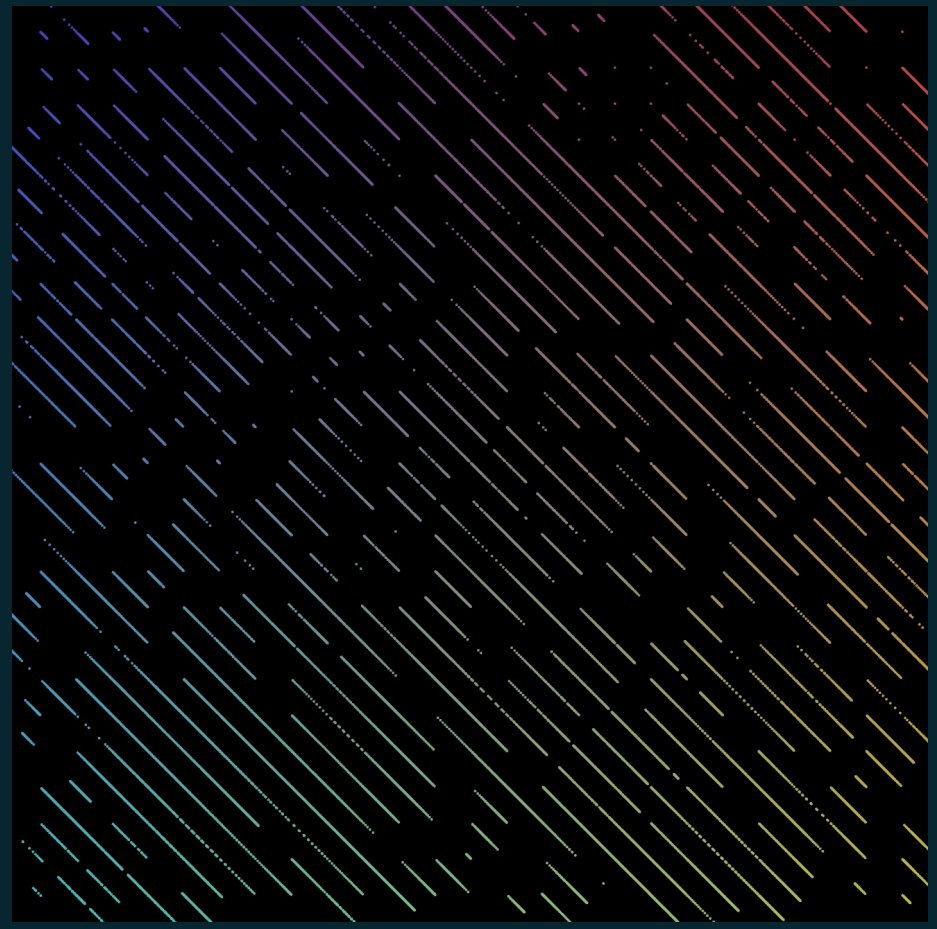
variation!("Position-dependent color");

```
let width = app.window_rect().w();
let height = app.window_rect().h();

for point in &model.points {
    let r = map_range(point.coordinates.x, -width, width, 0.0, 1.0);
    let g = map_range(point.coordinates.y, -height, height, 1.0, 0.0);
    let b = map_range(point.coordinates.x, -width, width, 1.0, 0.0);

    draw.ellipse()
        .color(rgb(r, g, b));
}
```



variation!("Different movements");

```
let noise_value_map =
    deg_to_rad(map_range(noise_value, -1.0, 1.0, -360.0, 360.0));

point.coordinates +=
    pt2(noise_value_map.tan(), -noise_value_map.tan())
```



RUSTFEST
GLOBAL 2021
Rust in Arts Edition

Thanks for your attention!

Francesco Cauteruccio, Ph.D.
f.cauteruccio@gmail.com
francescocauteruccio.info
twitter.com/finalfire
github.com/finalfire