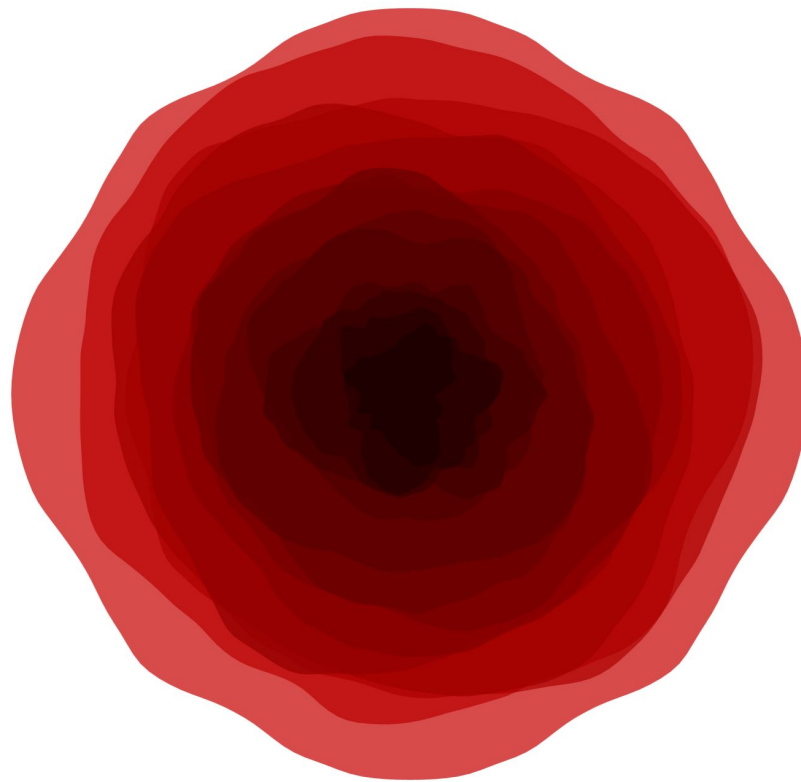


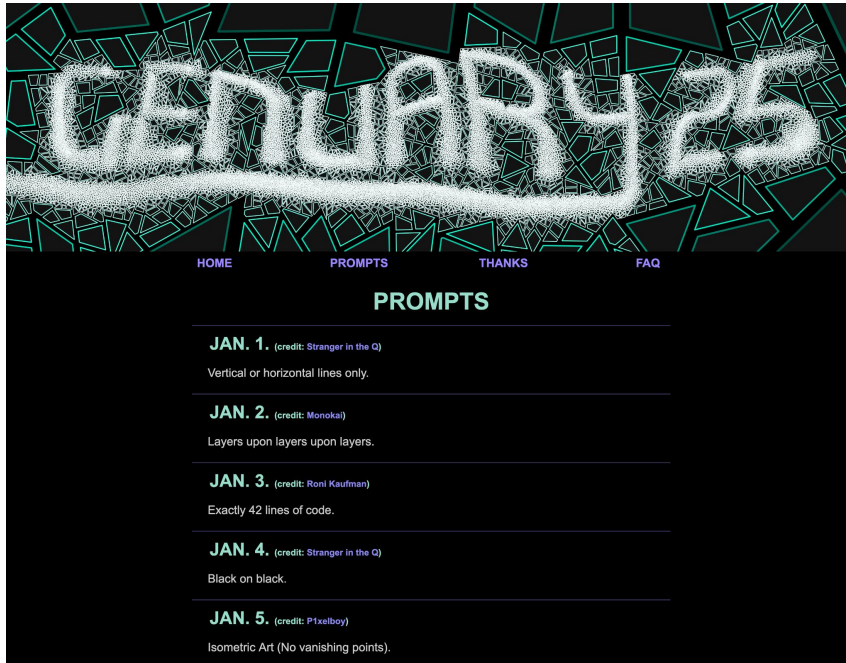
Genuary 2025: Lessons learned



ANYA PROSVETOVA

24 September 2025

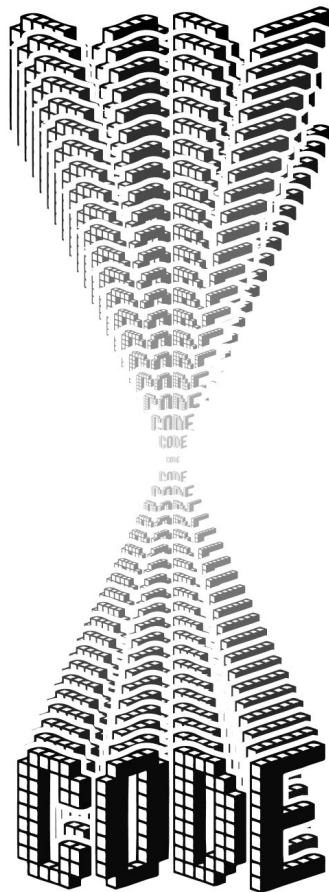
How it started



<https://genuary.art/>

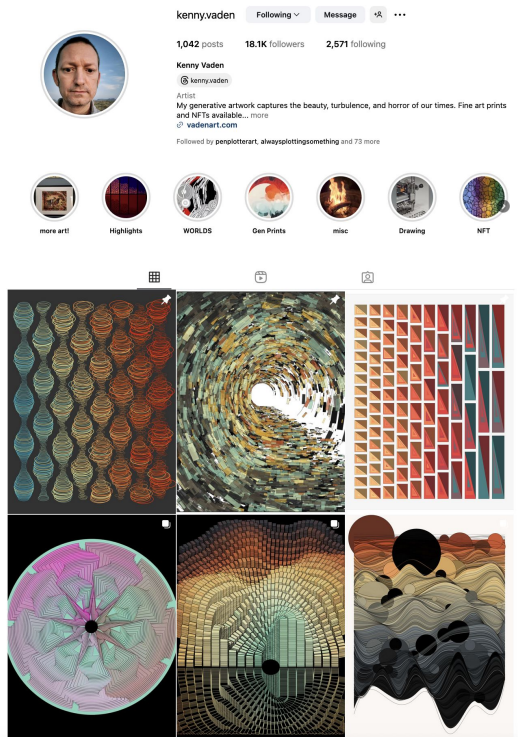
Choosing your **why**

- Learn
- Improve
- Experiment



January 3: Exactly 42 lines of code

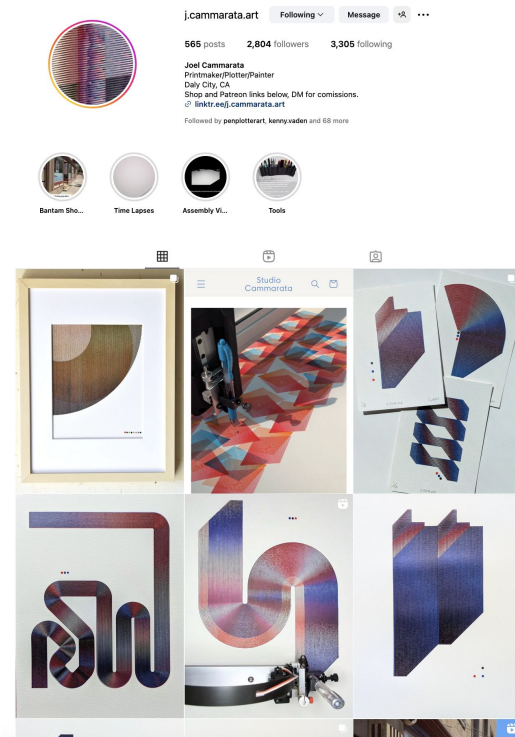
Inspiration: Creative coding community



kenny.vaden

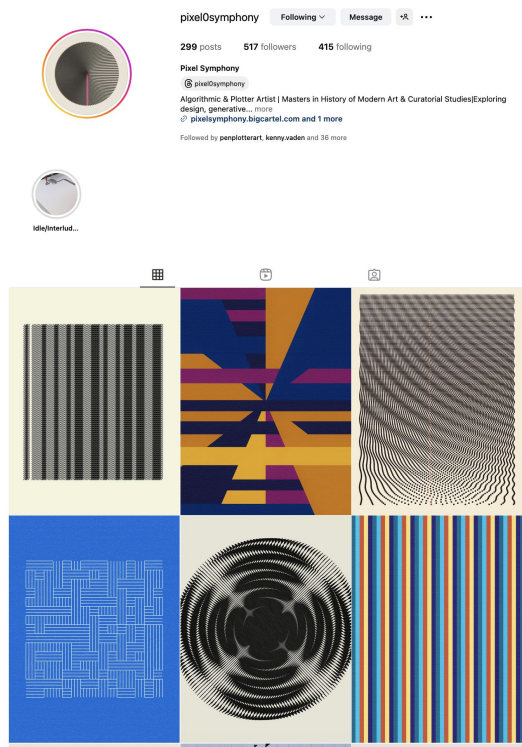


roni.kaufman.art

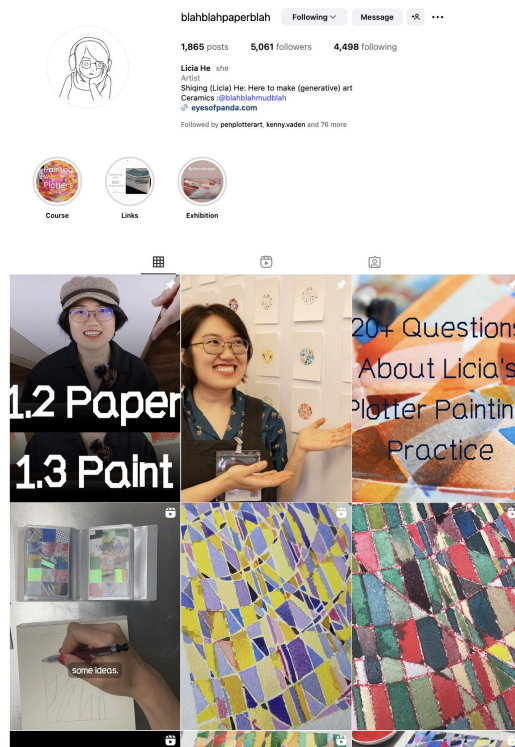


j.cammarata.art

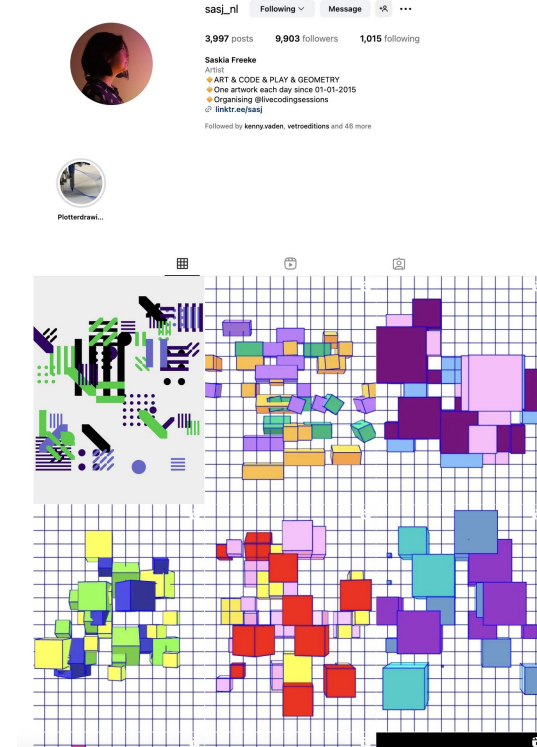
Inspiration: Creative coding community



pixel0symphony

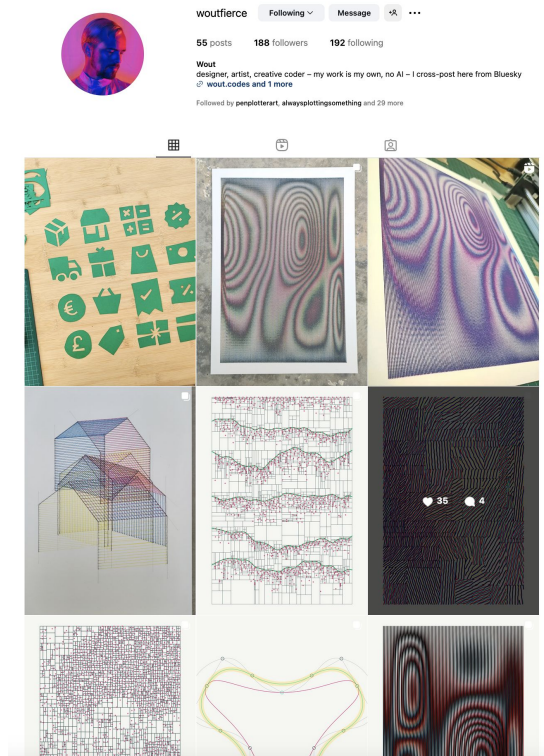


blahblahpaperblah

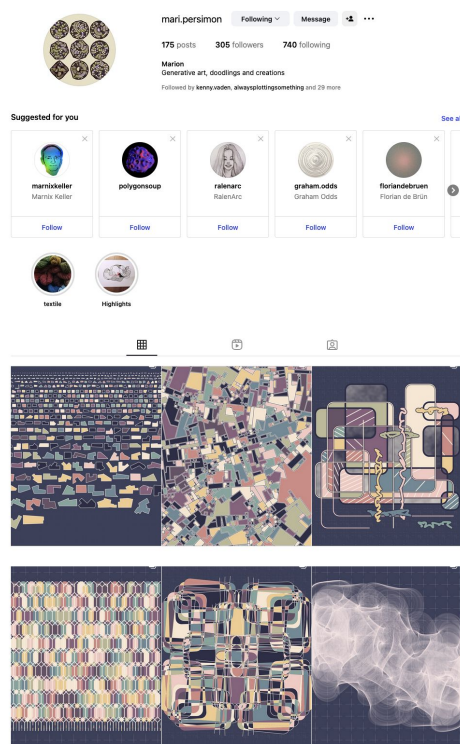


sasj_nl

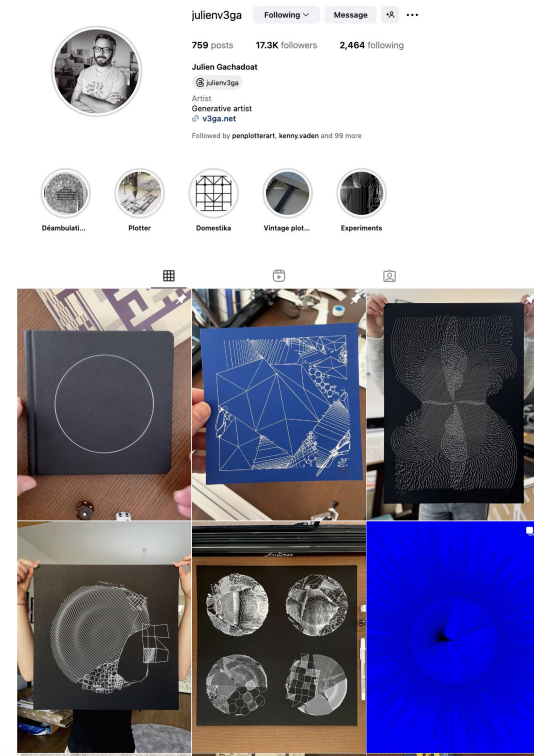
Inspiration: Creative coding community



woutfierce



mari.persimon



julienv3ga

Inspiration: code

- Bluesky
- GitHub
- [p5.js community](https://p5js.org/community)

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

English

Accessibility

Search

Community > Sketches

Sketches

Get inspired by sketches from the p5.js community.



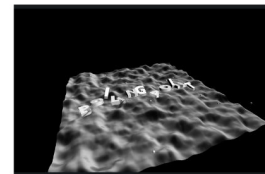
Feelings make us human
epibyte



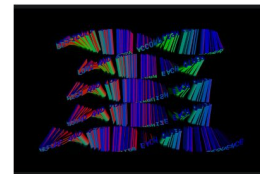
All you need is infinite love!
jpcnecmath



Emotional Echoes: Multilingual
Typography Experiment
Rikiya Okawa

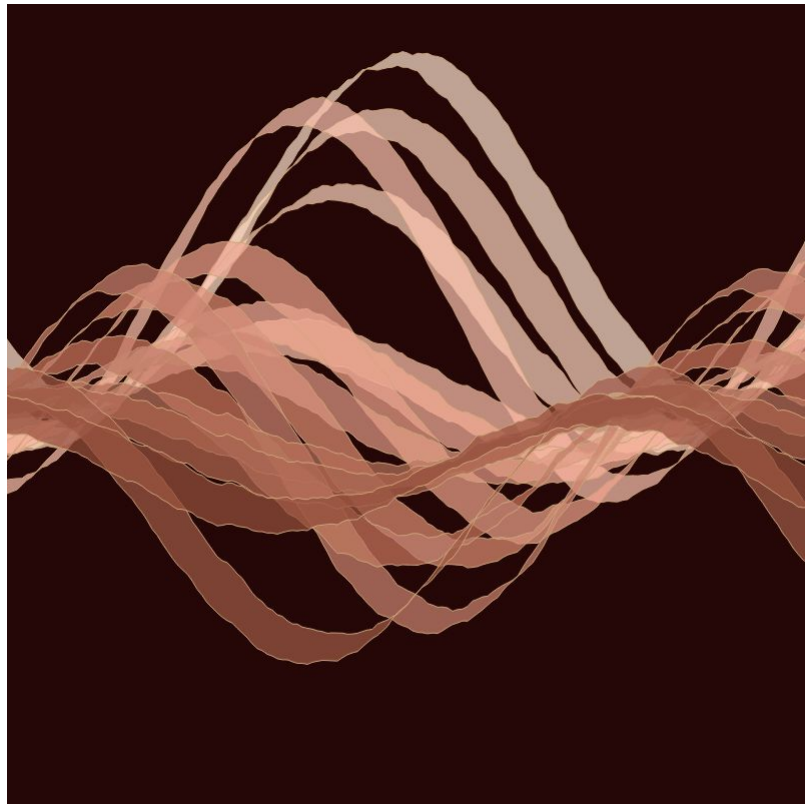


Boiling Point
Dave Pagurek



Creating your **process**

- Choose one technology
- Format constraints (image size, colour palette)
- Time constraints (for example, 1 hour per prompt)
- Daily / weekly coding



Keep going

- Track your progress
- Share your work on Bluesky or Instagram with common hashtags:

#genuity

#genuity{year}

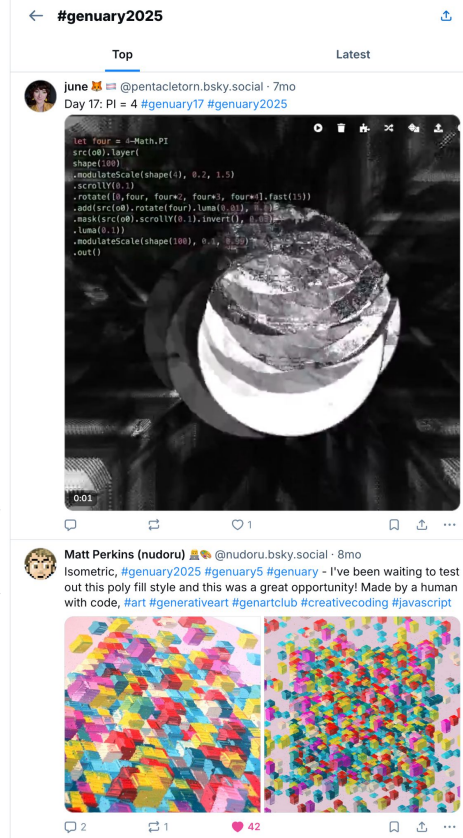
#genuity{date}

Genuary 2025

This repository contains generative art code created for [Genuary 2025](#) using daily prompts.

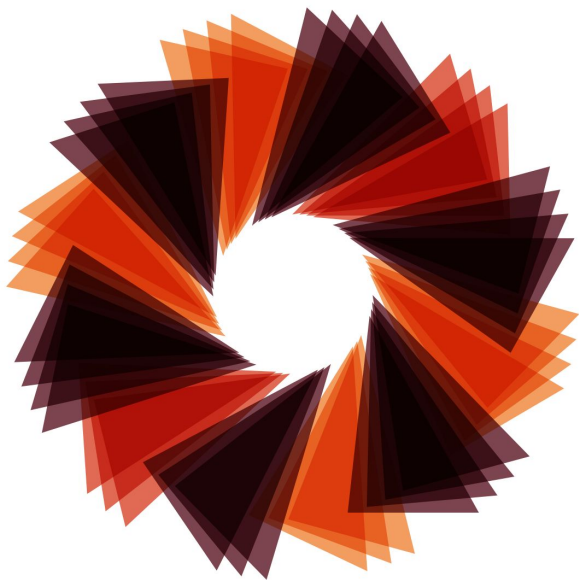
Prompts

- ✓ January 1. (credit: Stranger in the Q) Vertical or horizontal lines only.
- ✓ January 2. (credit: Monokai) Layers upon layers upon layers.
- ✓ January 3. (credit: Roni Kaufman) Exactly 42 lines of code.
- ✓ January 4. (credit: Stranger in the Q) Black on black.
- ✓ January 5. (credit: P1xelboy) Isometric Art (No vanishing points).
- ✓ January 6. (credit: Jonathan Barbeau) Make a landscape using only primitive shapes.
- ✓ January 7. (credit: Camille Roux) Use software that is not intended to create art or images.
- ✓ January 8. (credit: Piter Pasma) Draw one million of something.
- ✓ January 9. (credit: Piter Pasma) The textile design patterns of public transport seating.
- ✓ January 10. (credit: Darien Brito) You can only use TAU in your code, no other number allowed. $\text{TAU} = 2 * \pi = 6.2831853...$



Keep **going**: reuse code from other days

January 2: Layers upon layers upon layers



January 13: Triangles and nothing else

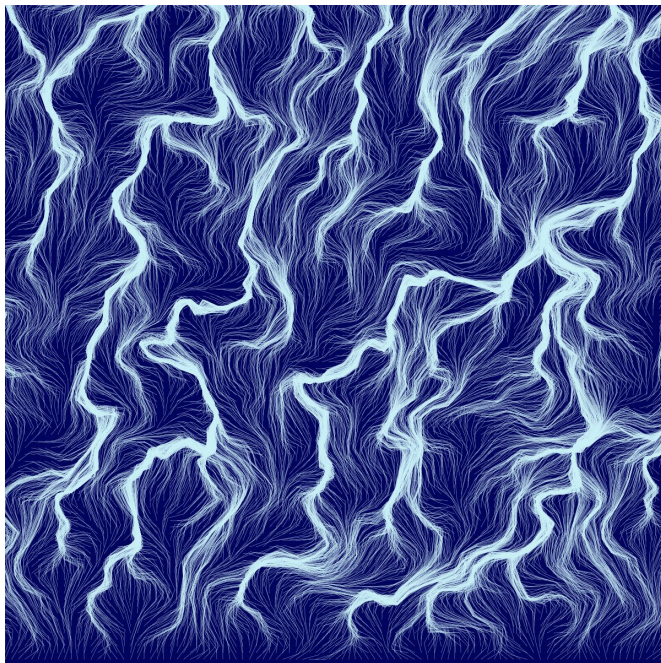


Keep **going**: reuse code from other days

January 18: What does wind look like?

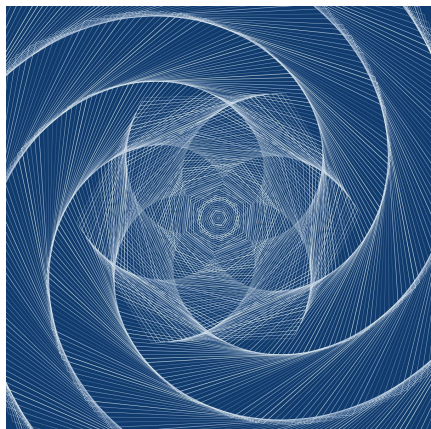


January 30: Abstract map. Not to be confused with AbstractMap

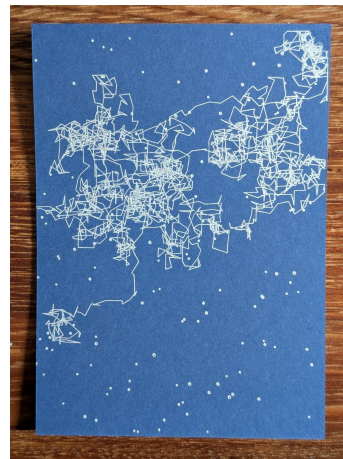
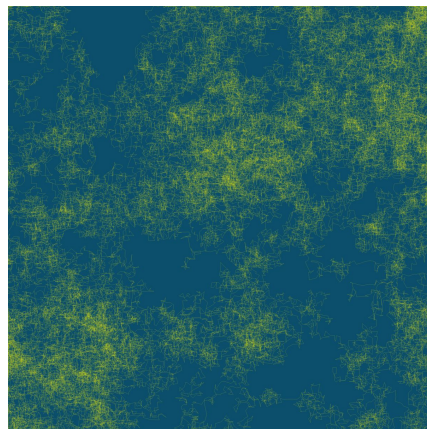


Keep **going**: print your designs

January 26: Symmetry

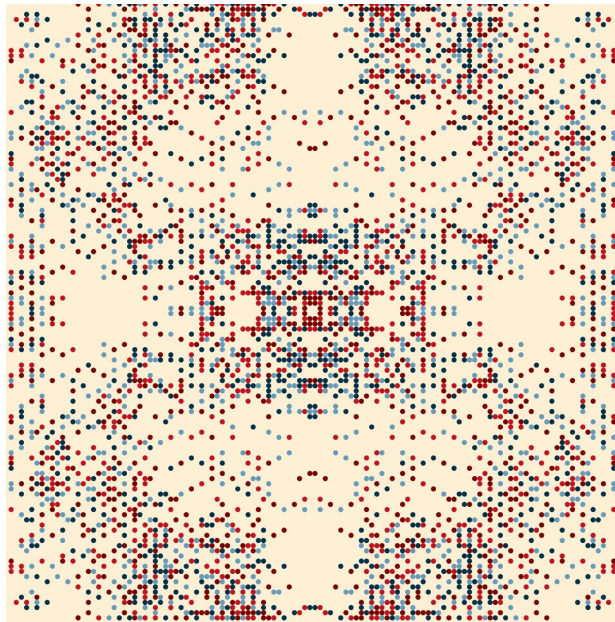


January 25: One line that may or may not intersect itself



Keep **going**: use of AI

- Explain code from others or documentation
- Find inspiration from generated images (Midjourney, etc.)



January 15: Design a rug

I have a p5.js script that generates a pattern made of dots. However, it's very slow and I want to understand how to speed up the generation of the pattern. Here is the current version:

```
...  
  
// Number of columns and rows in the grid  
let cols;  
  
Show more ▾
```

Answer

To speed up your p5.js pattern-generation script **without altering the visual output or core design logic**, here are practical optimizations that directly target your performance bottleneck:

1. Avoid Full Grid Redraw on Every Frame

Currently, the script redraws the entire grid every frame by:

- Calling `background('#FDFD05')` in `draw()`
- Repainting **every** circle in nested `for` loops, even if only a few cells change.

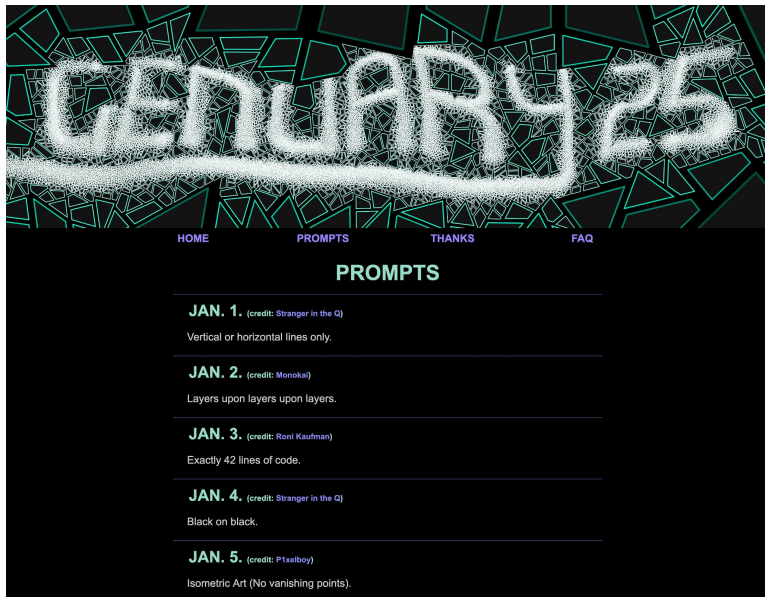
Optimization:

- **Draw only the four new/mirrored circles in each iteration** (not the full grid each frame).
- Set the background and initially draw the blank grid in `setup()`. Then, within `draw()`, just draw the newly colored circles.

How to Implement

```
javascript  
function setup() {  
  createCanvas(600, 600, SVG);  
  cols = width / size;  
  rows = height / size;  
  x = floor(random(cols / 2 + margin, cols / 2 + margin));  
  y = floor(random(rows / 2 + margin, rows / 2 + margin));  
  // Initialize grid array  
  grid = Array.from([length: cols], () => Array.from([length: rows], () => color));  
  background('#FDFD05');  
  // Draw the blank grid once  
  for (let i = 0; i < cols; i++) {  
    for (let j = 0; j < rows; j++) {  
      // ...  
    }  
  }  
}
```

How it started



<https://genuary.art/>

How it's going



Genuary 2025: My Journey to Learn p5.js

Feb 1, 2025

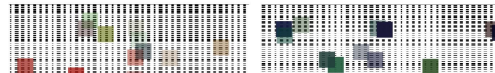
This year I finally did something that I've been meaning to do for a couple of years now: I completed all Genuary prompts within the month of January.

Genuary is a creative coding challenge that happens every January. During this month participants follow daily prompts to create artworks using code. The challenge doesn't specify which programming language (or any other software) participants should use, so I chose to complete the challenges using p5.js JavaScript library. I've seen many great designs created using this library and have been meaning to learn it for a while. This year I decided to persevere with the challenges, and I'm so happy I did! I learnt a lot about the library, different algorithms that can be used to create impressive generative art, and got inspired to continue working with it to develop my creative ideas.

This blog is a collection of all generative art pieces I created for Genuary 2025 using daily prompts. I plan to explore some of these visual ideas further in the coming months, so stay tuned for more posts about p5.js and how you can use it to create beautiful images. I also [share my work on Bluesky](#).

January 1. (credit: Stranger in the Q)

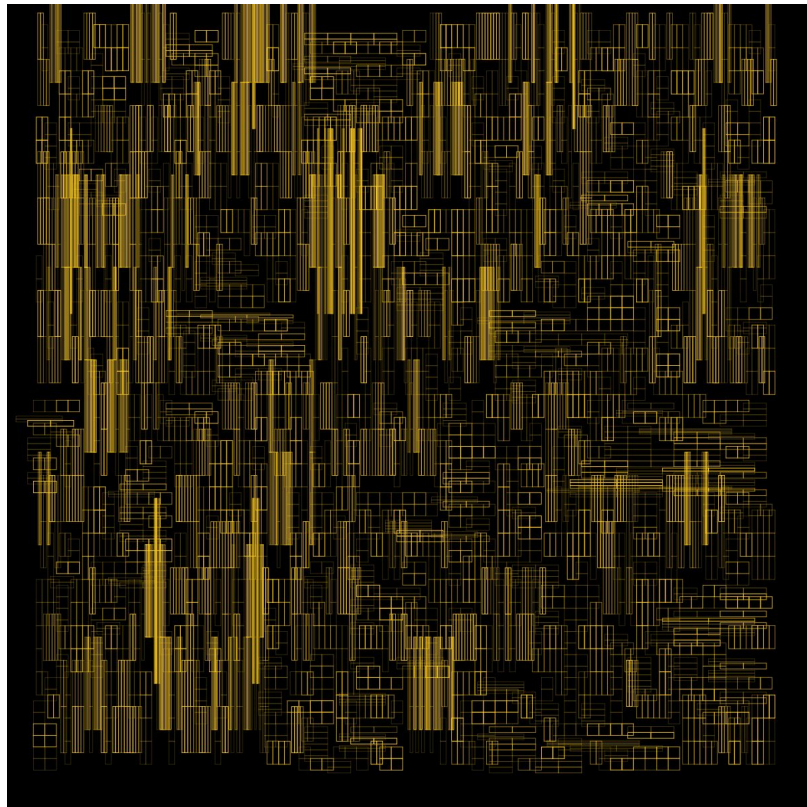
Vertical or horizontal lines only



<https://www.anvalitica.dev/blog/2025-02-01-genuary-2025-collection>

My plan for **Genuary 2026**

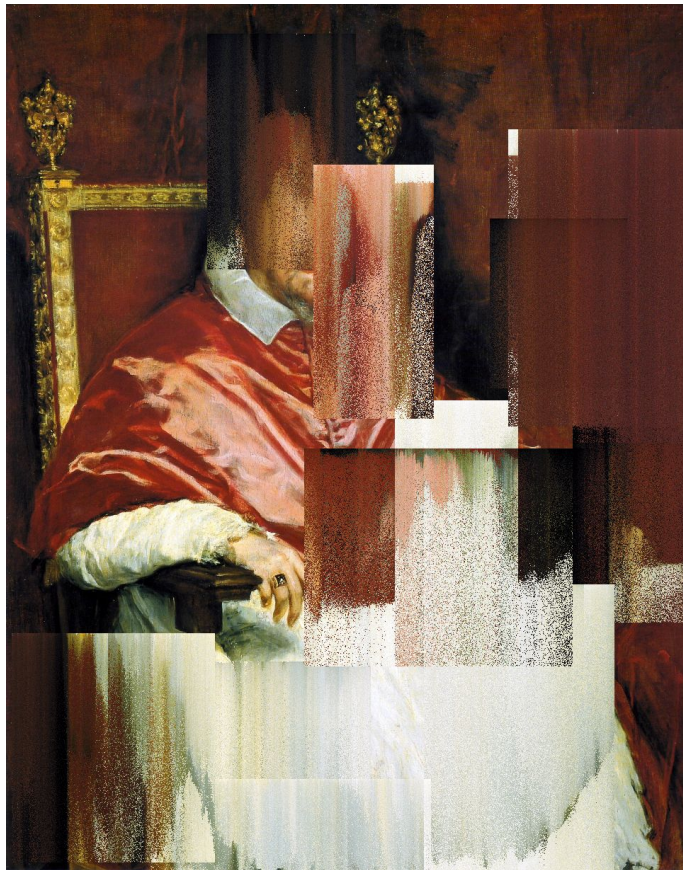
- Set format constraint: size, colour palette
- Think about how I can group different prompts / reuse techniques
- Focus on how I can design for plotter



January 12: Subdivision

Useful resources

- [Genuary](#) website
- [p5.js reference](#)
- [Repositories on GitHub](#)
- [Coding train](#) tutorials
- [Patt Vira](#) tutorials



January 31: Pixel sorting

Questions?

Get in touch:



anyalitica.dev

Thank you!

