

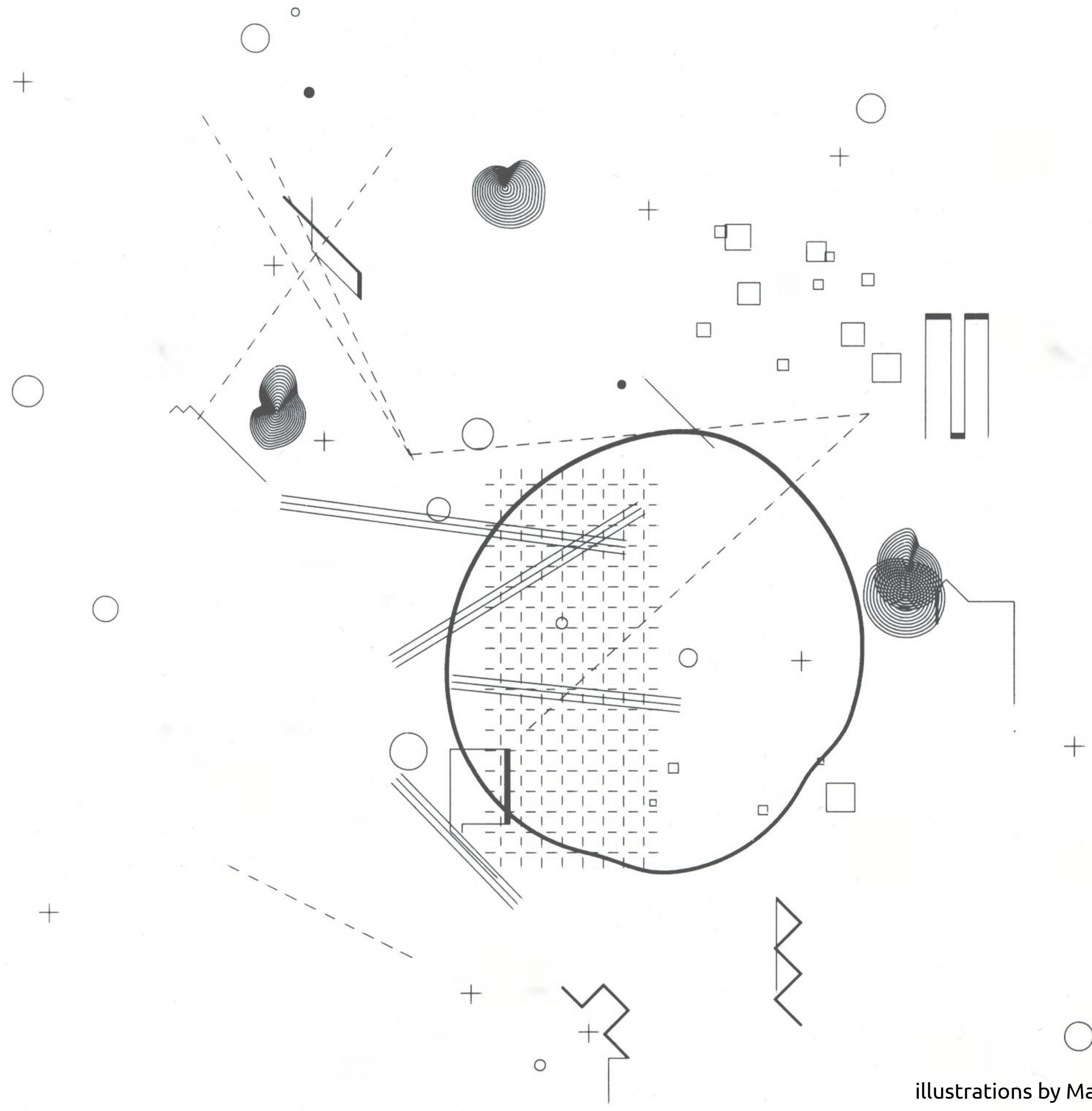
GRAPHICS CODING - JS/P5

Term 1

November 2019

Lior Ben-Gai

<https://soogbet.github.io/ShenkarCC/>



illustrations by Manfred Mohr

GOALS

- 1. Improve our visual communication skills through computer programming.**
- 2. Improve our computer programming skills through visual communication.**

COURSE TOPICS

1. Language, abstraction and procedural drawing
2. Variables, Functions, Conditions and Loops
3. Interaction and animation
4. Data structures and visualization
5. Generative design methods
6. Introduction to 3D structures, image processing and HTML

ADMIN

Schedule:

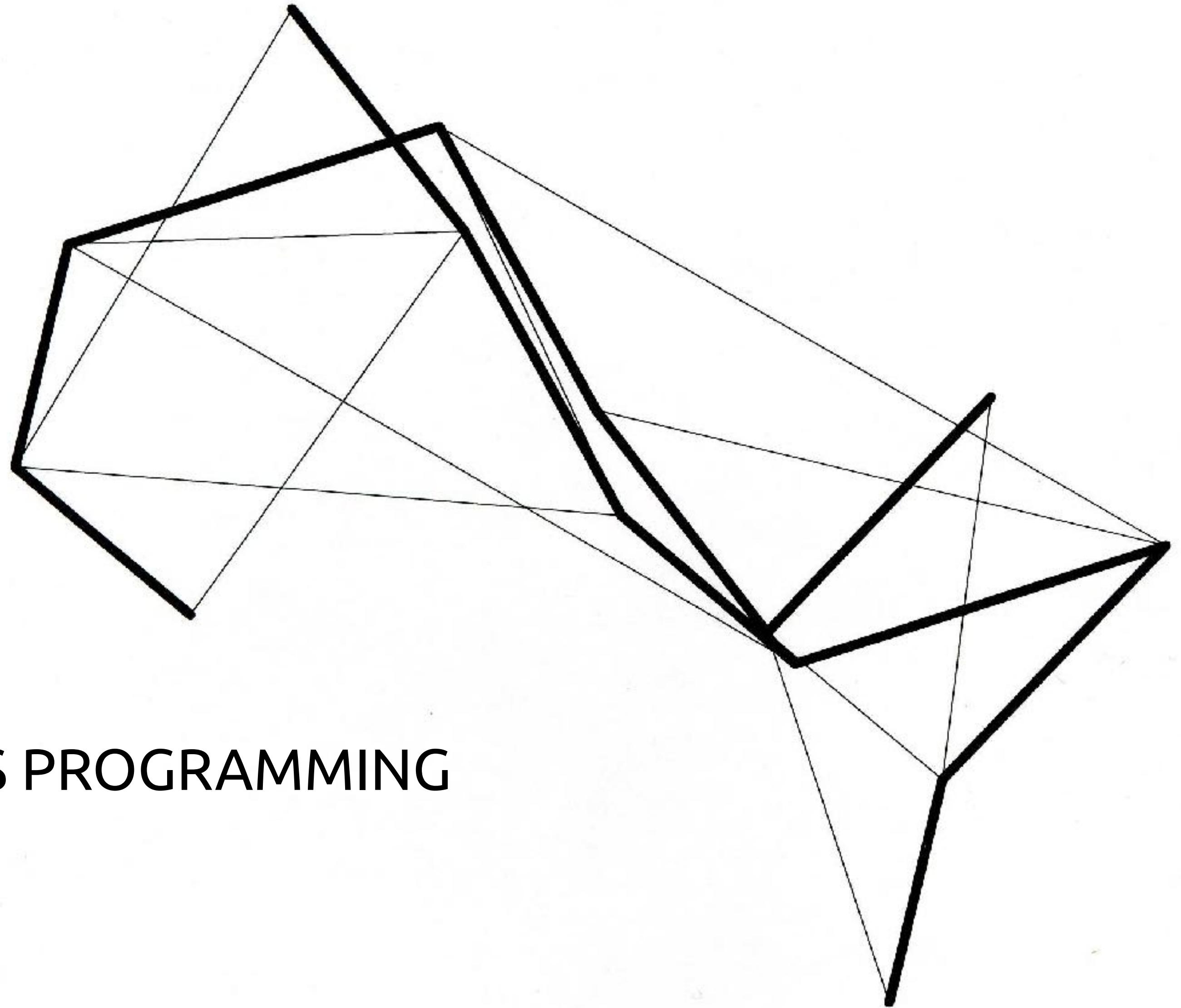
Thursdays 13:00 - 16:00 (Mandatory)

30.10.19 >> 30.01.20

Marking:

13 weekly assignments: 55% (4 pt each + 3 pt bonus)

1 final project - 45% (topic and date TBD)



INTRODUCTION TO GRAPHICS PROGRAMMING

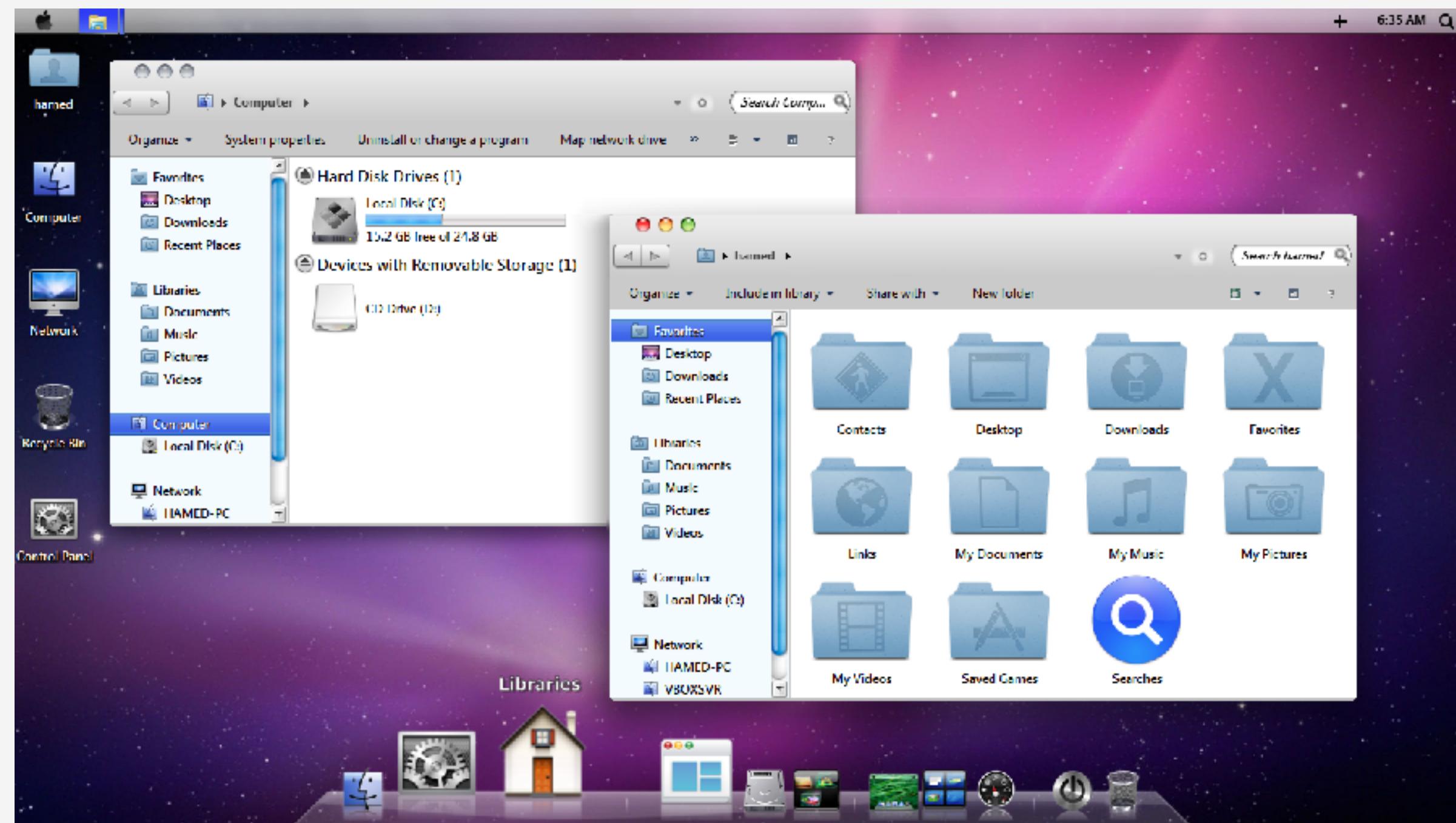
Language / Abstraction / Procedural drawing

WHAT IS COMPUTER PROGRAMMING ANYWAY?

IS THIS COMPUTER PROGRAMMING?



IS THIS COMPUTER PROGRAMMING?

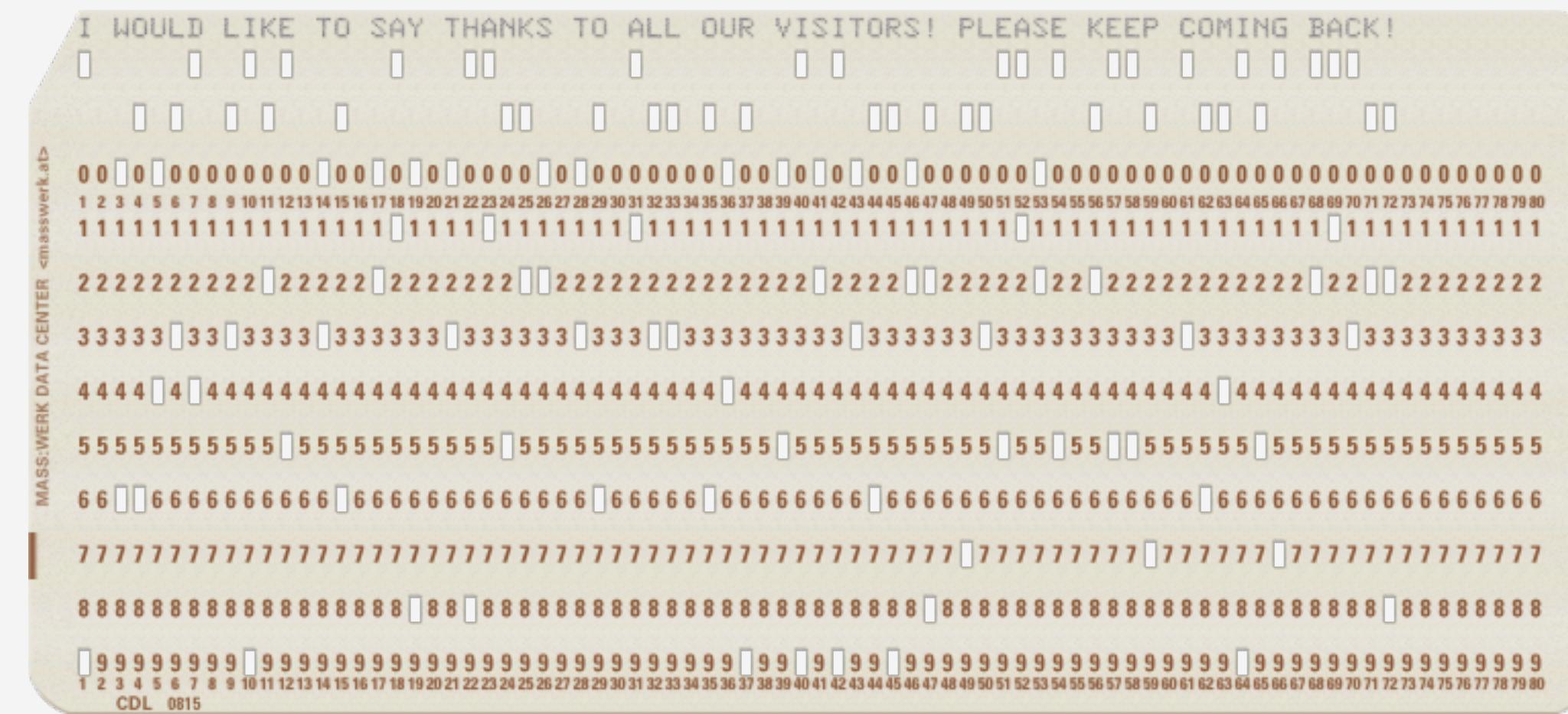


IS THIS COMPUTER PROGRAMMING?

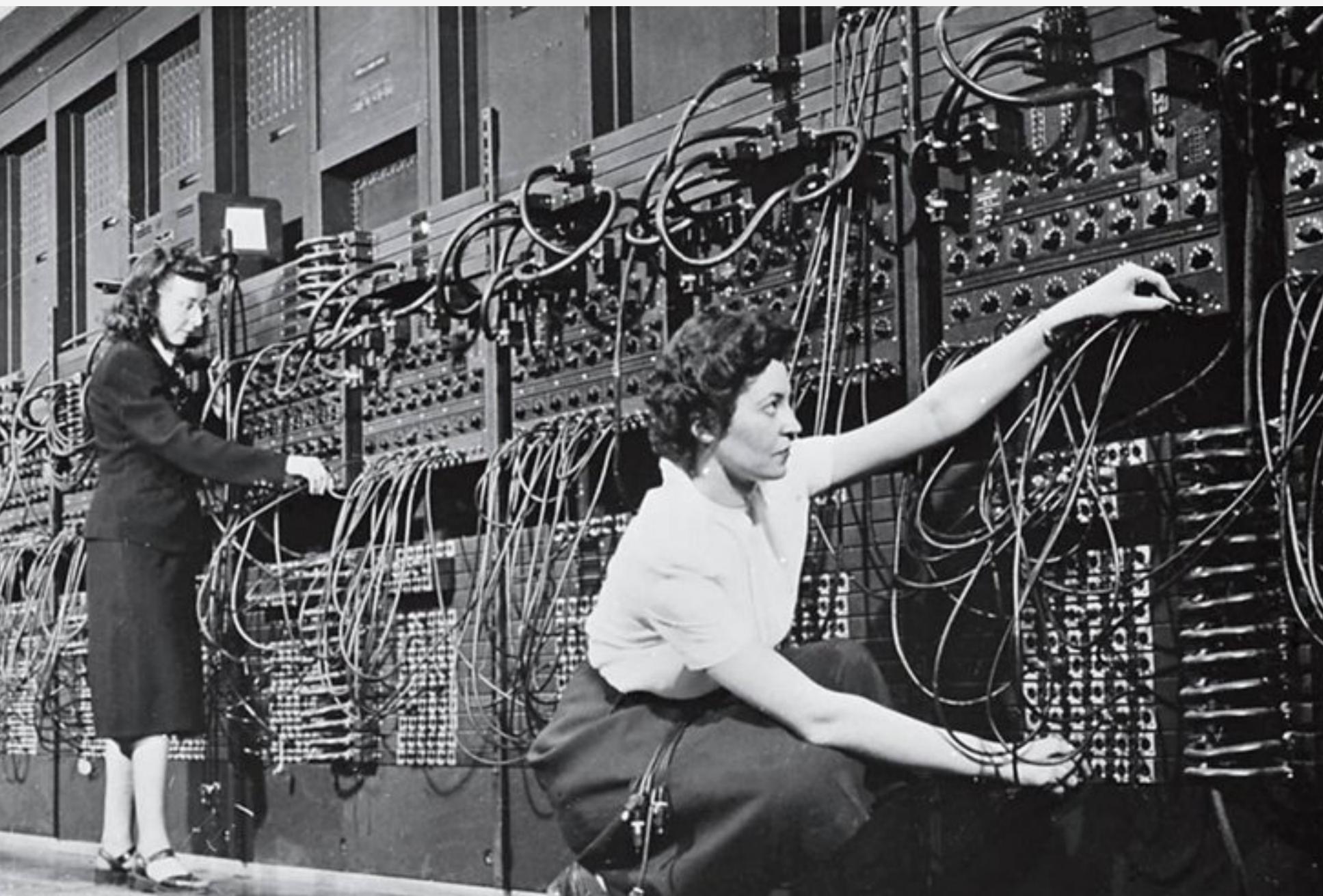
The screenshot shows a code editor window with multiple tabs at the top. The active tab is "GPUComputationRenderer.js". The code itself is a JavaScript class definition for "GPUComputationRenderer". It initializes a scene, camera, and mesh, and provides methods for adding variables with specific shaders and initial textures.

```
99
100 function GPUComputationRenderer( sizeX, sizeY, renderer ) {
101
102     this.variables = [];
103
104     this.currentTextureIndex = 0;
105
106     var scene = new THREE.Scene();
107
108     var camera = new THREE.Camera();
109     camera.position.z = 1;
110
111     var passThruUniforms = {
112         texture: { value: null }
113     };
114
115
116     var passThruShader = createShaderMaterial( getPassThroughFragmentShader(), passThruUniforms );
117
118     var mesh = new THREE.Mesh( new THREE.PlaneBufferGeometry( 2, 2 ), passThruShader );
119     scene.add( mesh );
120
121
122     this.addVariable = function( variableName, computeFragmentShader, initialValueTexture ) {
123
124         var material = this.createShaderMaterial( computeFragmentShader );
125
126         var variable = {
127             name: variableName,
128             initialValueTexture: initialValueTexture,
129             material: material,
130             dependencies: null,
131             renderTargets: [],
132             wrapS: null,
133             wrapT: null,
134             minFilter: THREE.NearestFilter,
135             magFilter: THREE.NearestFilter
136         };
137
138         this.variables.push( variable );
139     };
140 }
```

IS THIS COMPUTER PROGRAMMING?



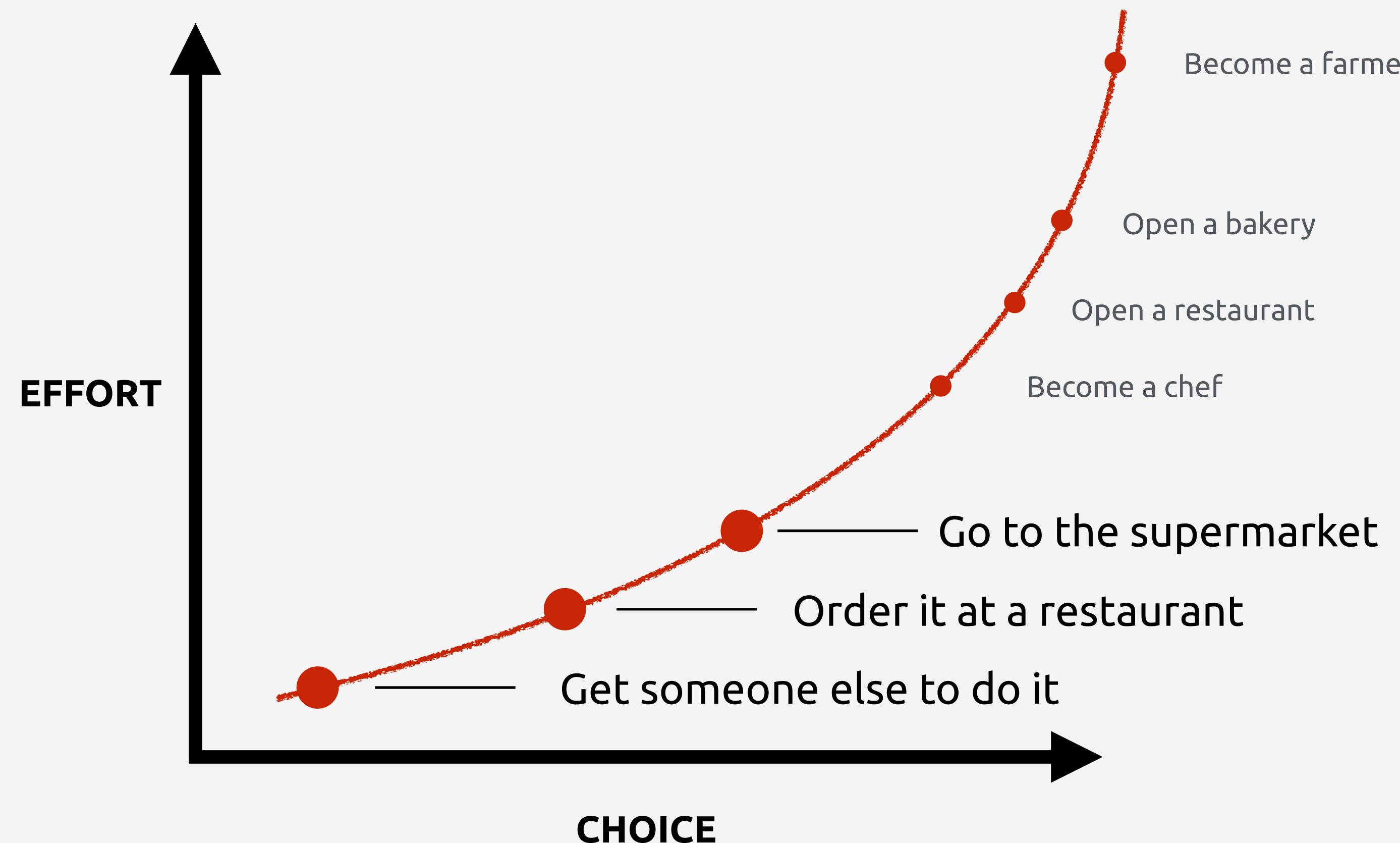
IS THIS COMPUTER PROGRAMMING?



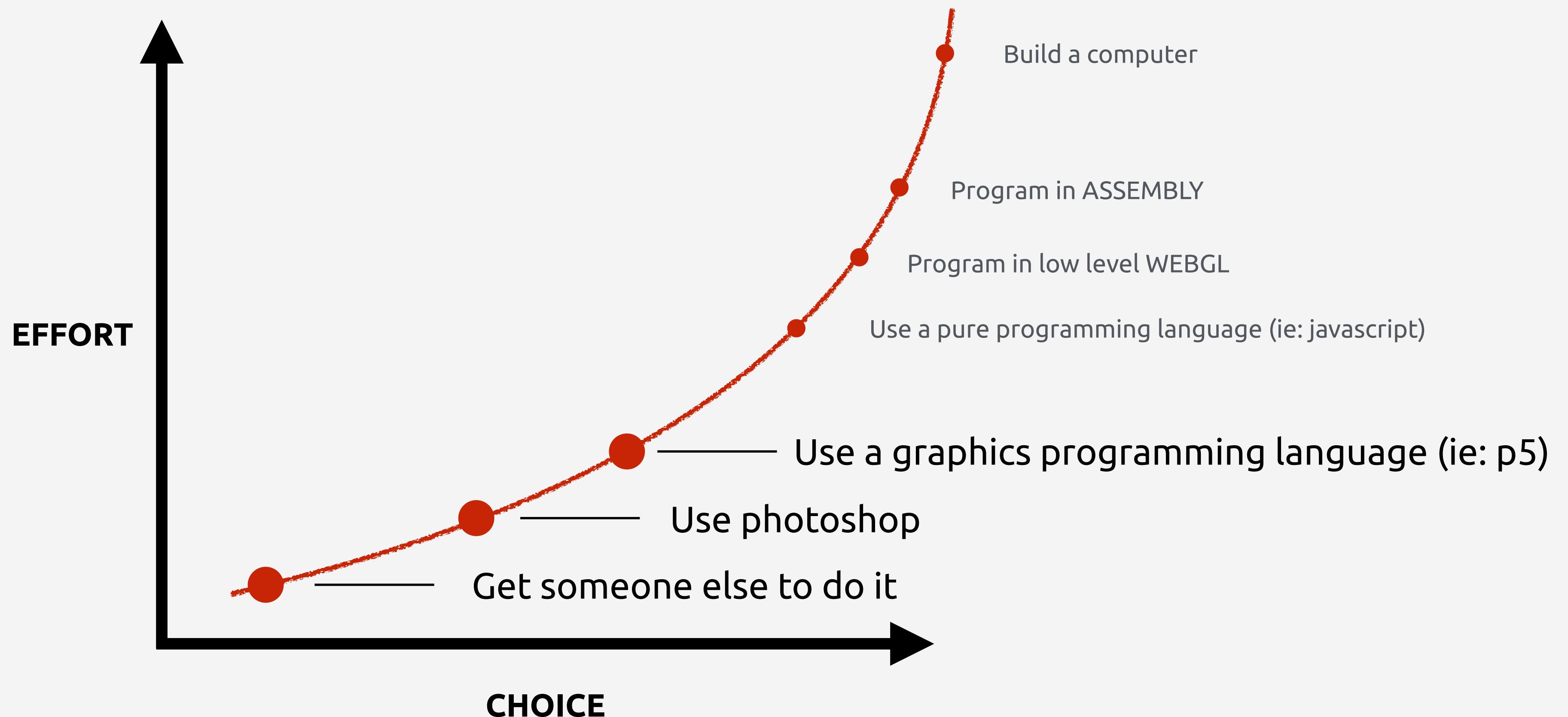
IS THIS COMPUTER PROGRAMMING?



MAKING A SANDWICH

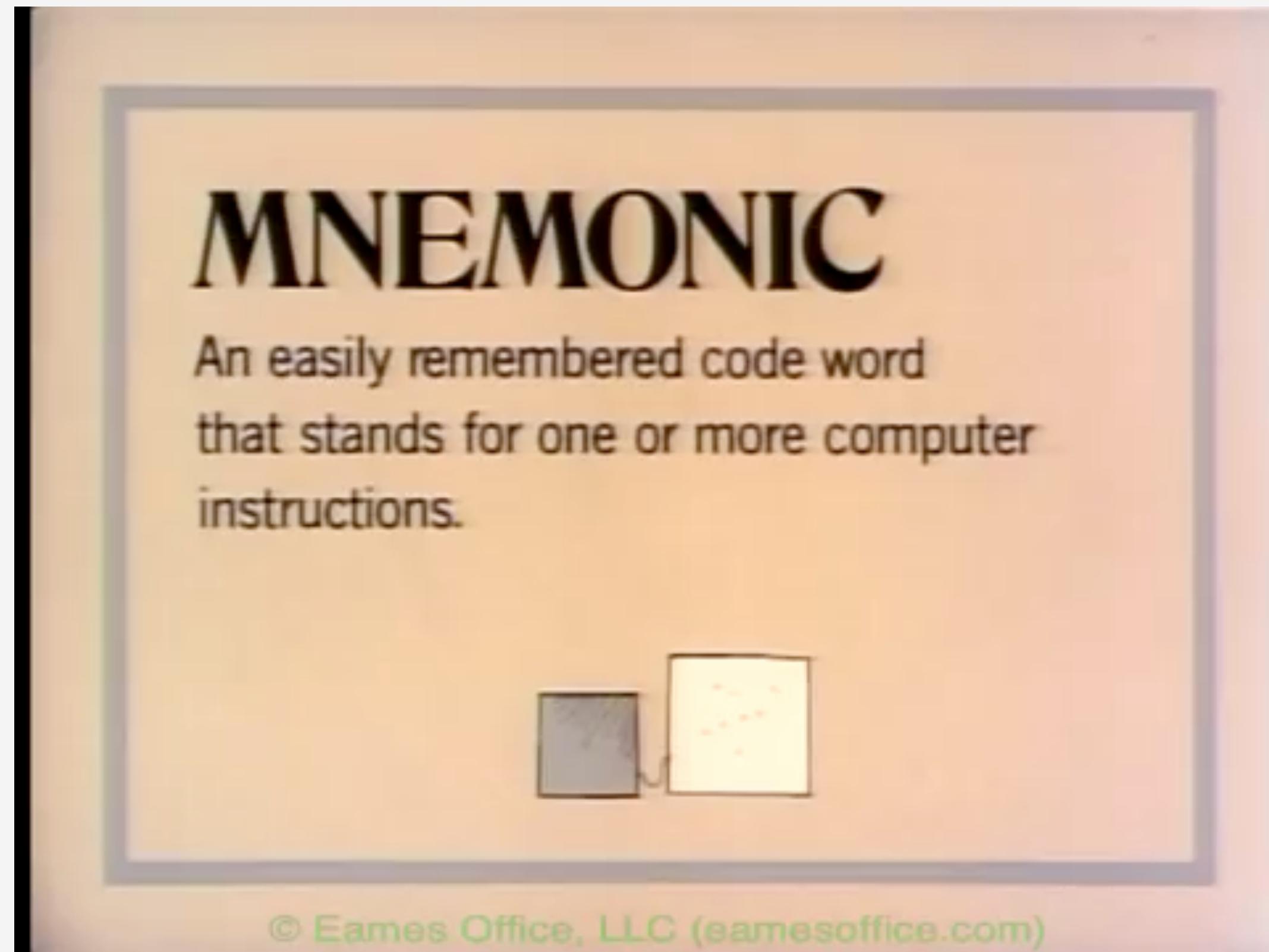


CREATE A DIGITAL ARTWORK



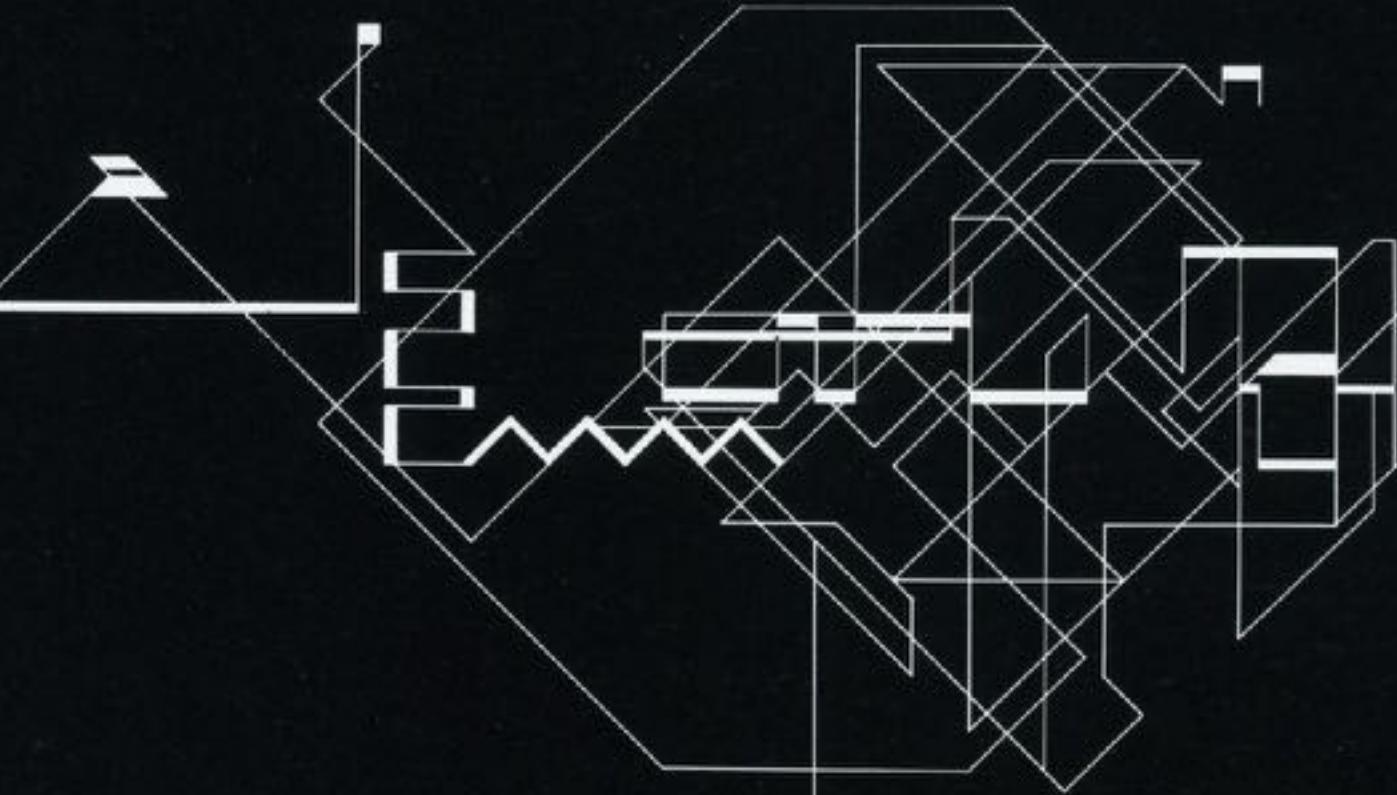
A COMPUTER GLOSSARY

Charles and Ray Eames 1966

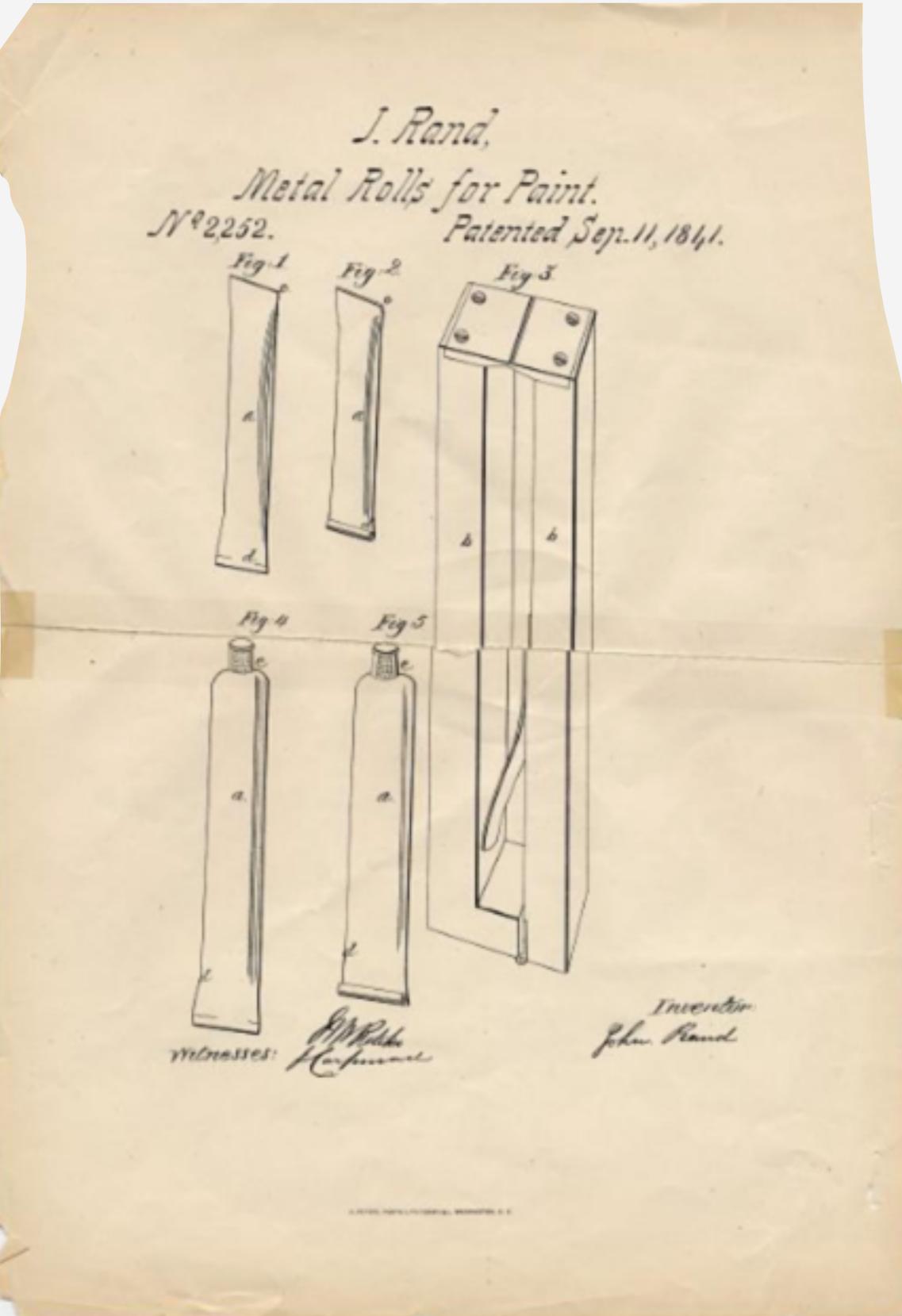


<https://www.youtube.com/watch?v=elgX6sPOqCY>

BUT WHAT DOES THAT HAVE TO DO WITH DESIGN?



ART AND TECHNOLOGY

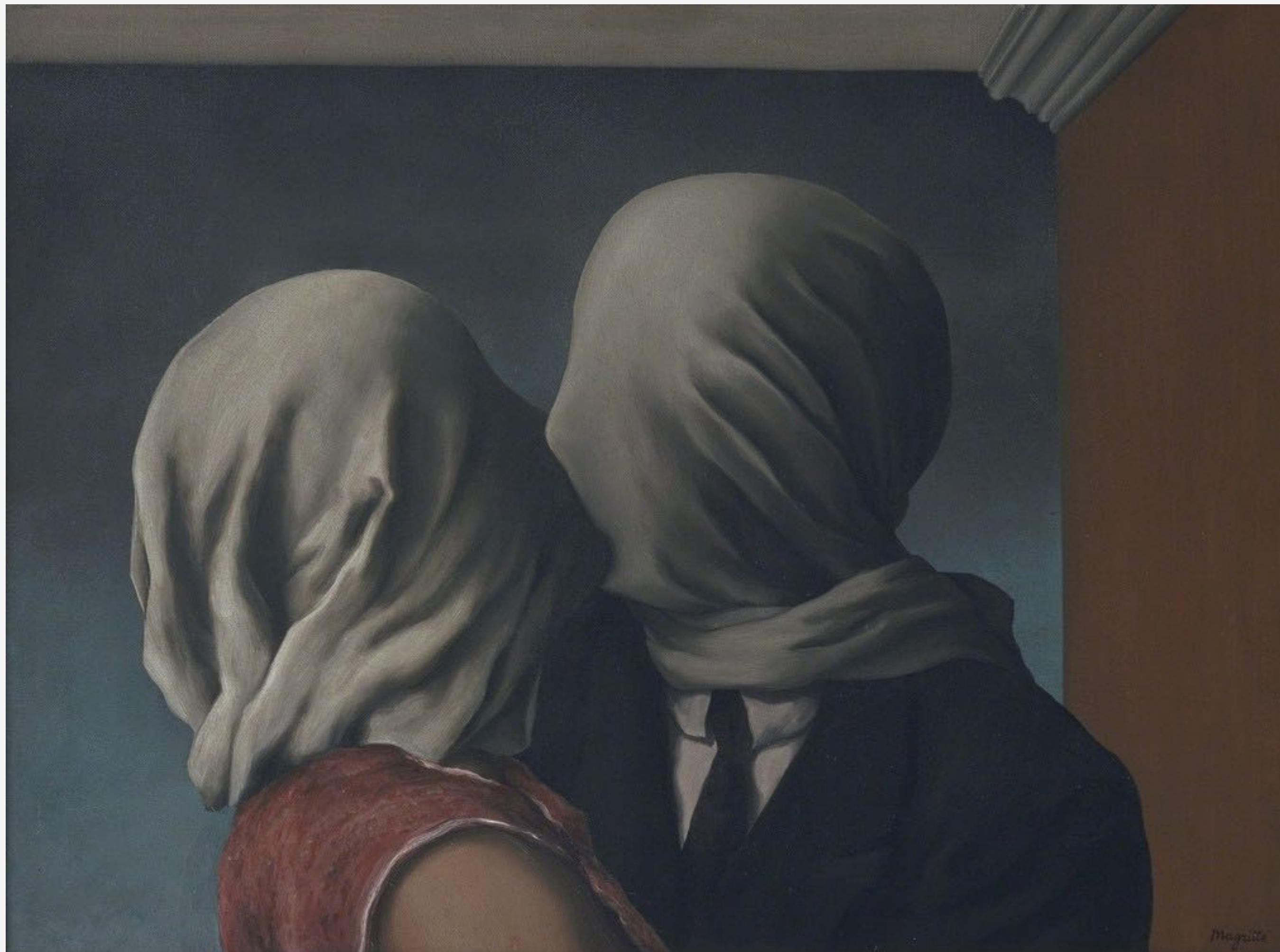


John Goffe Rand, Collapsible metal tube



Claude Monet, Impression, soleil levant

VISUAL STORY TELLING



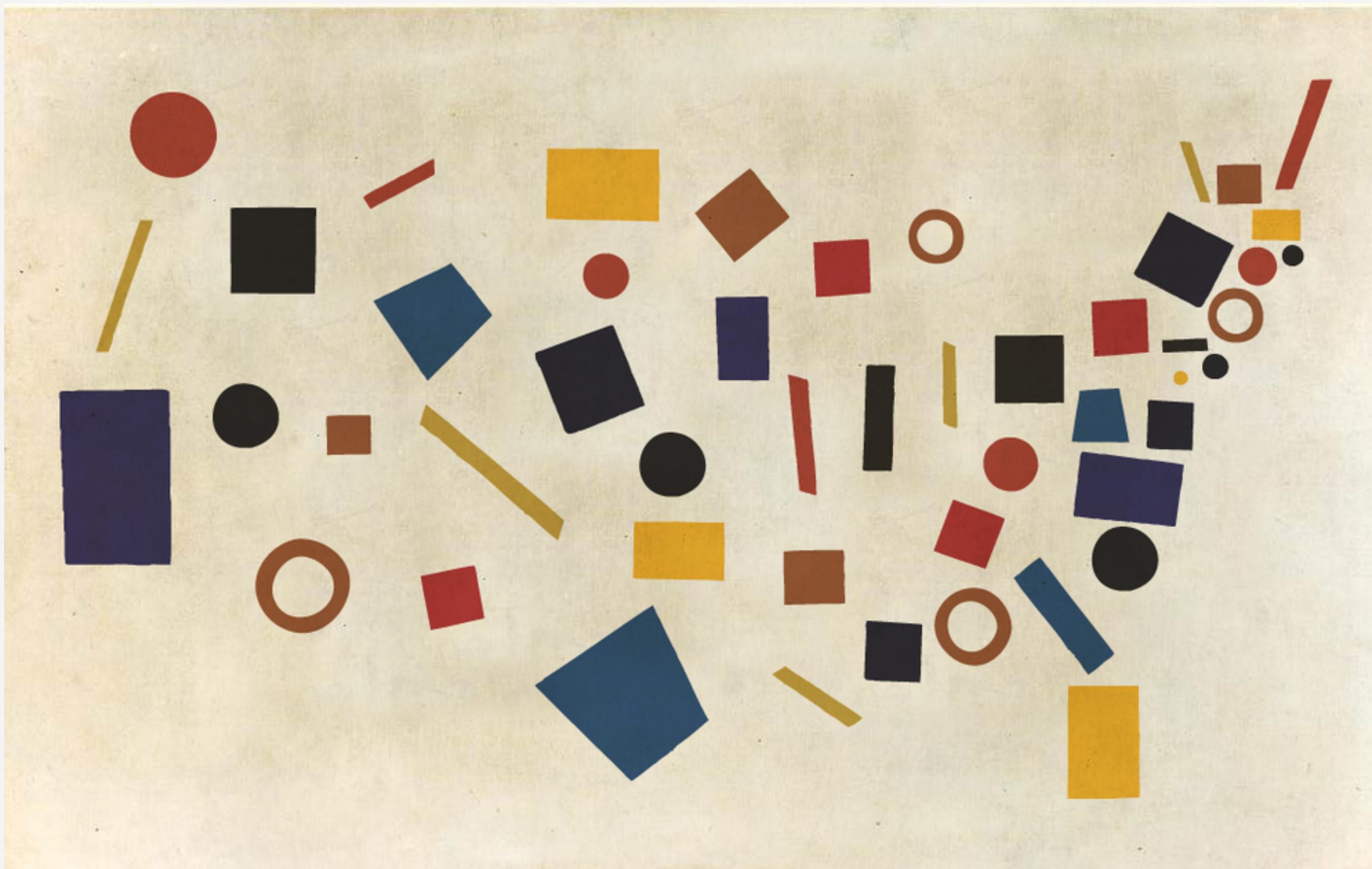
Rene Magritte



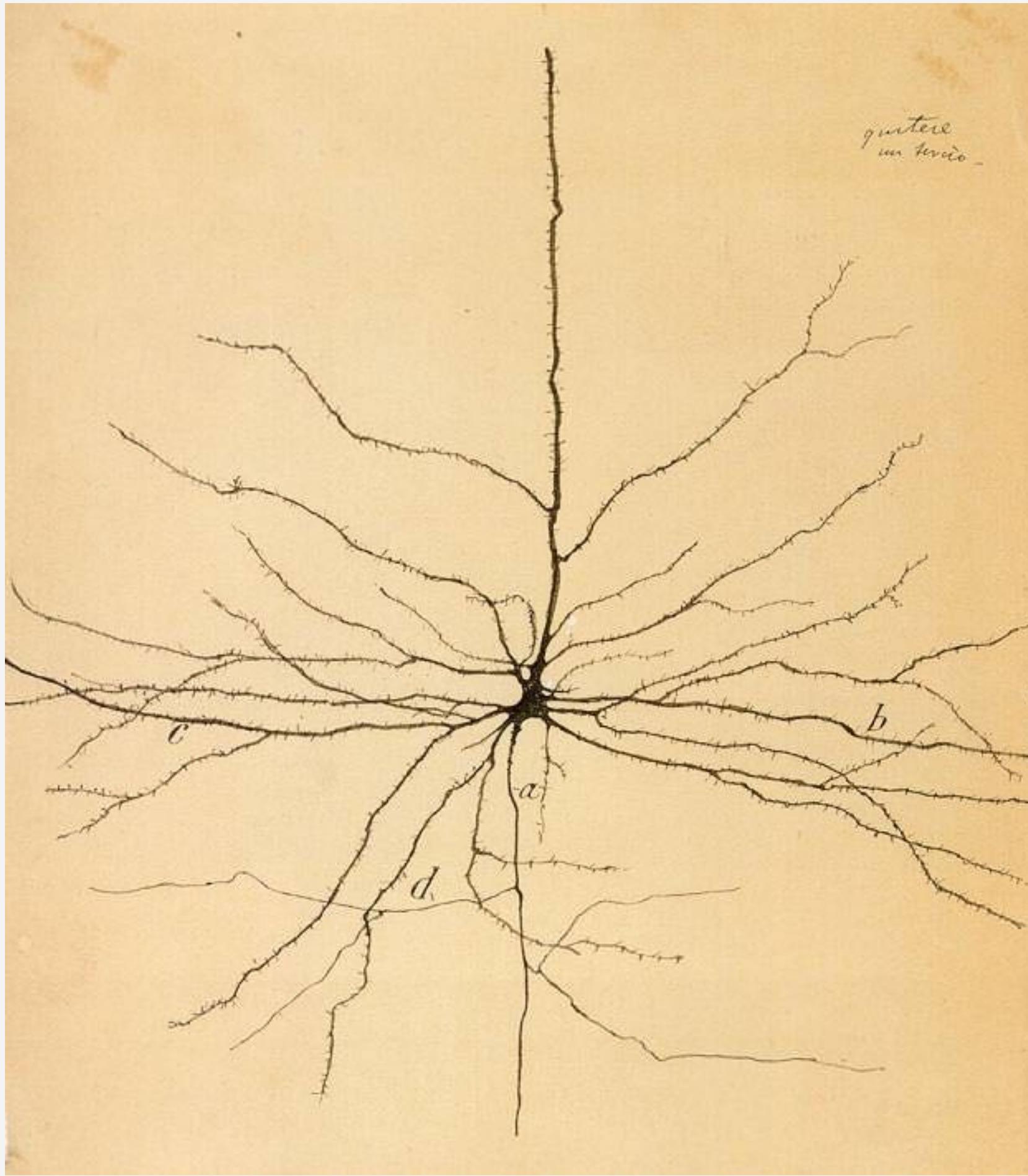
Anselm Kiefer



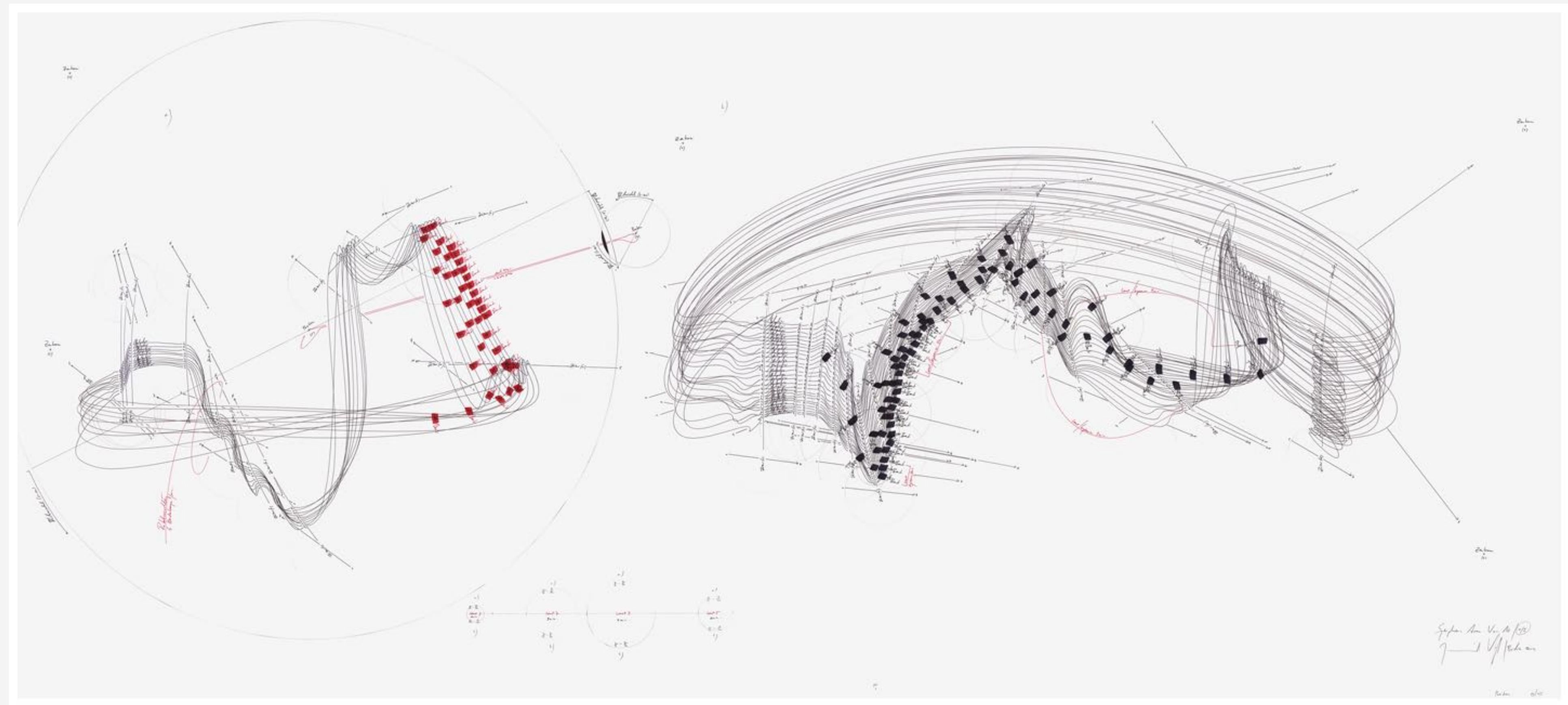
Luigi Serafini



Kazimir Malevich



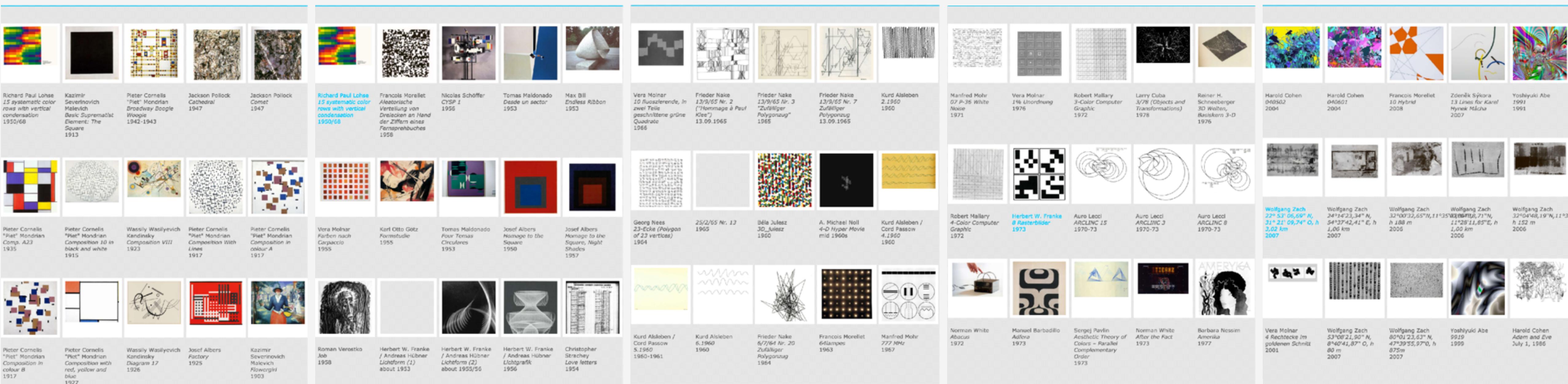
Santiago Ramón y Cajal



Jorinde Voigt

BRIEF HISTORY OF COMPUTER ART

Computational and algorithmic art during the 20th century



Before 1950

1950 - 1960

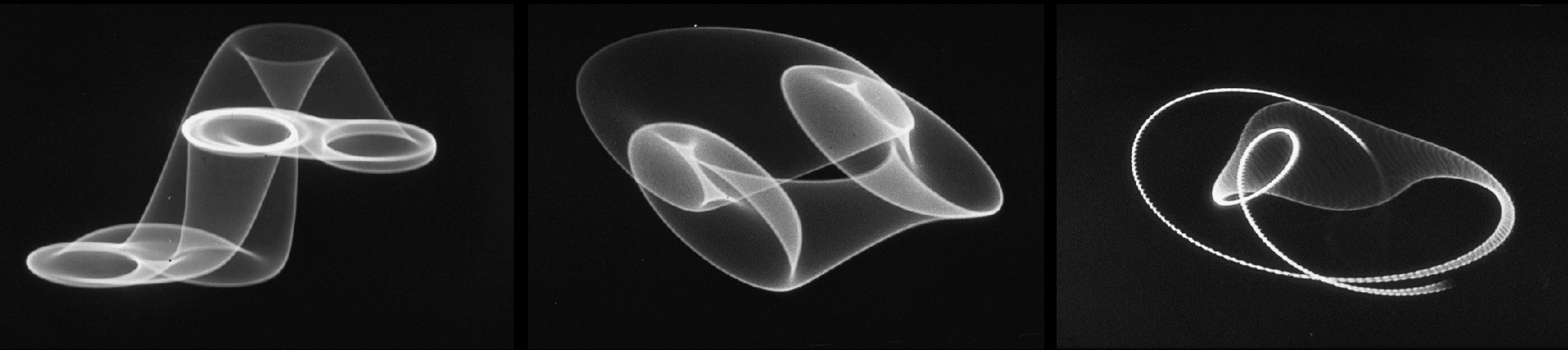
1960 - 1970

1970 - 1980

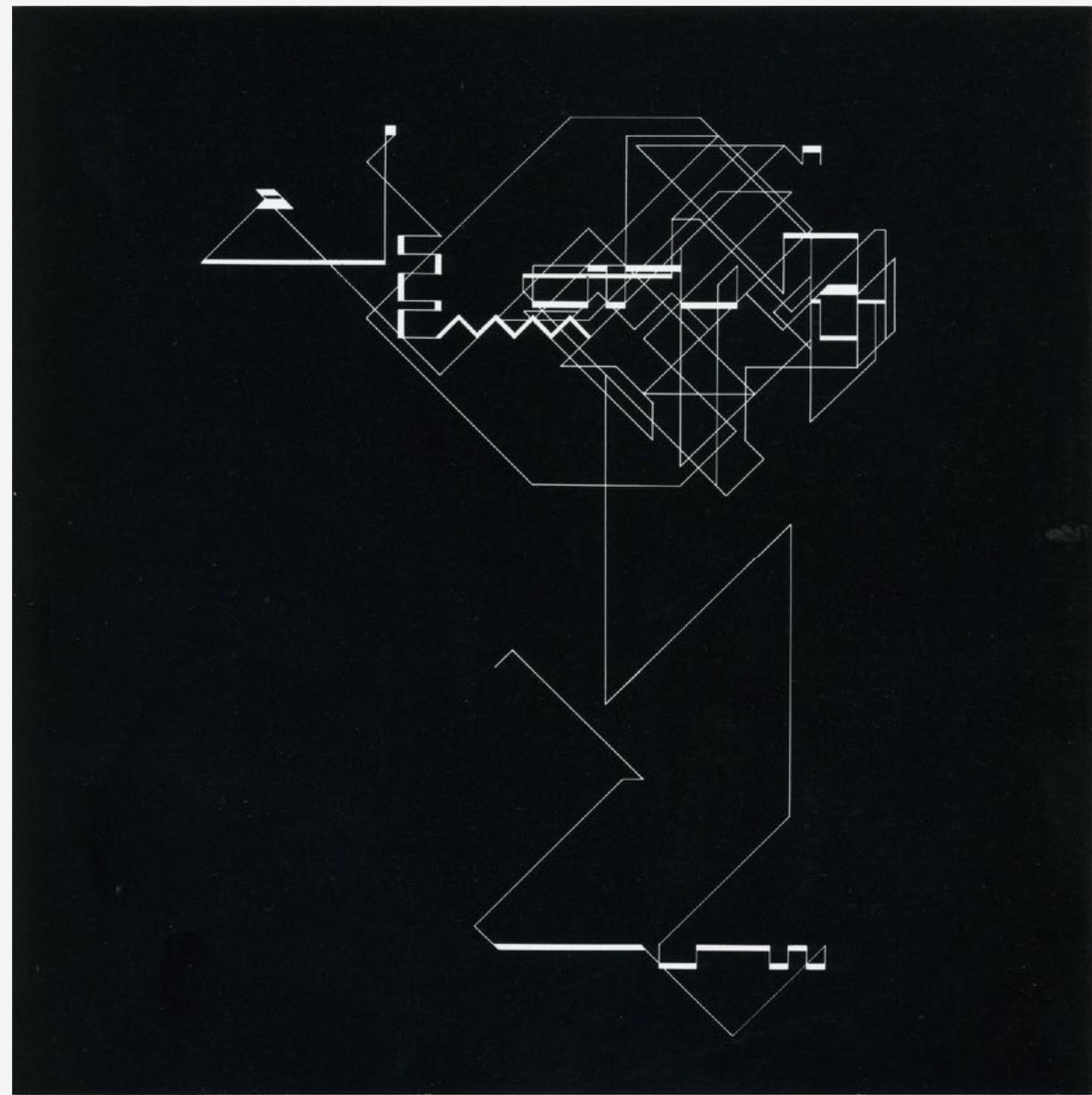
After 1980

<http://dada.compart-bremen.de>

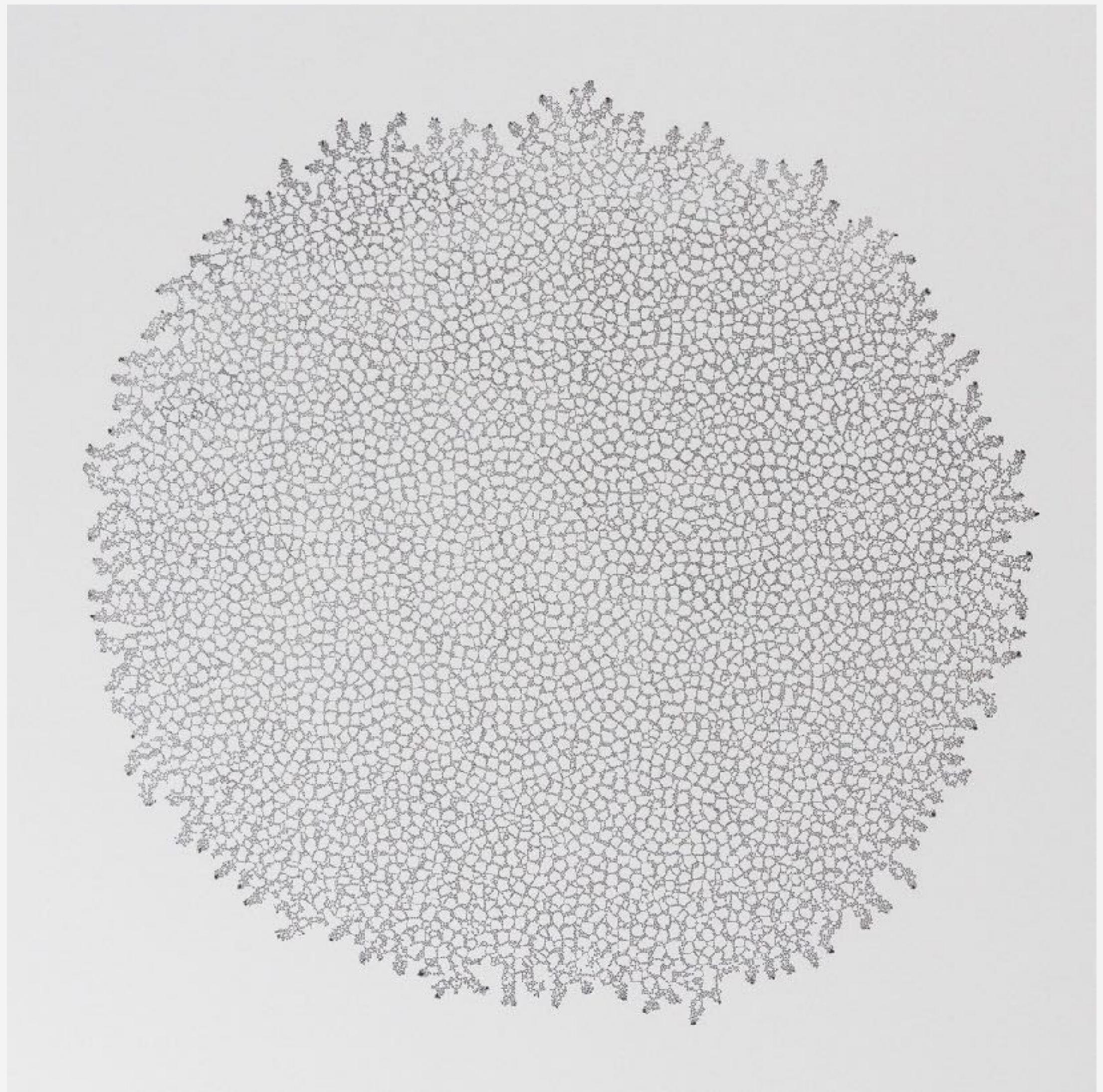
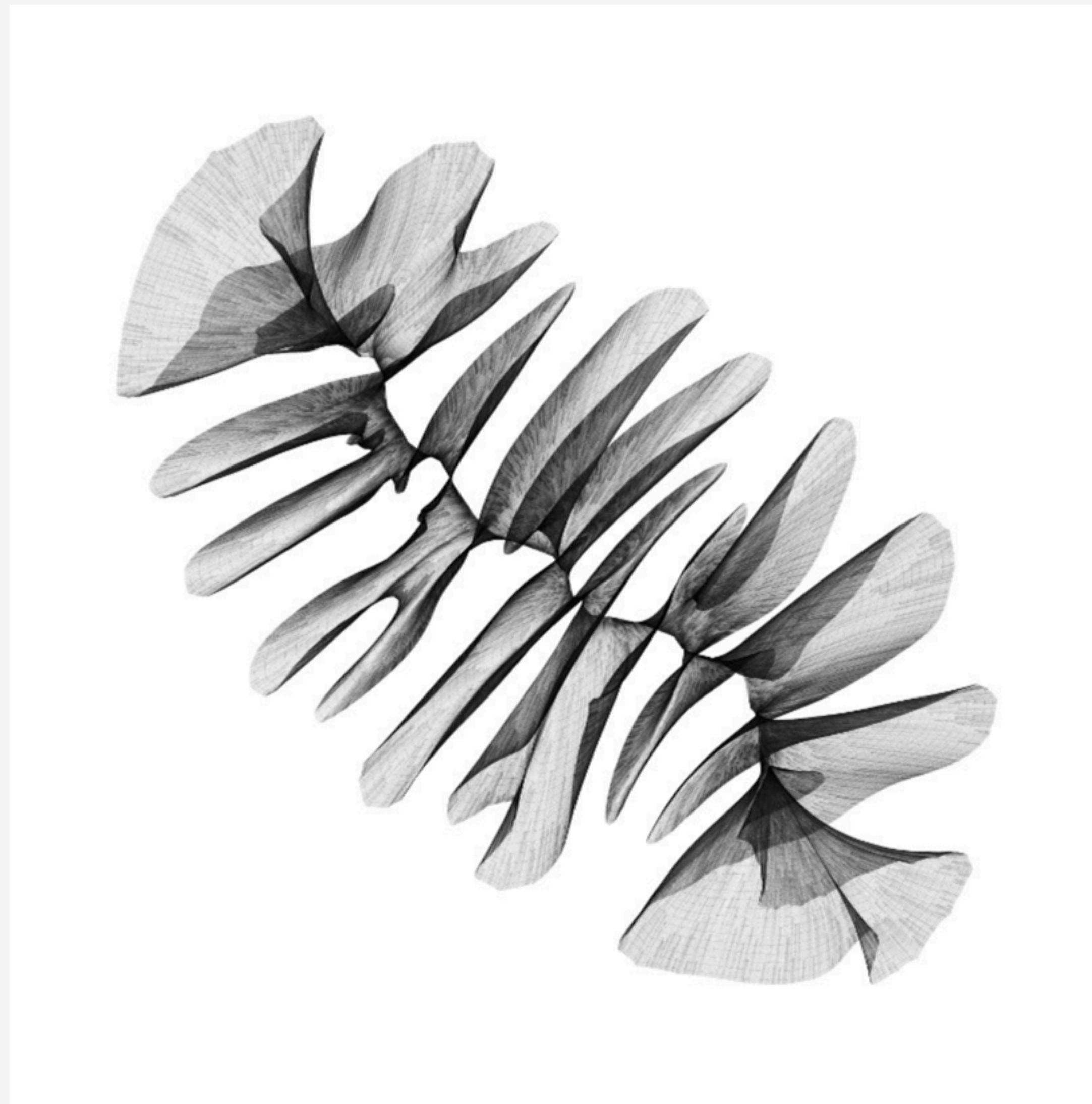
<http://digitalartmuseum.org/history>



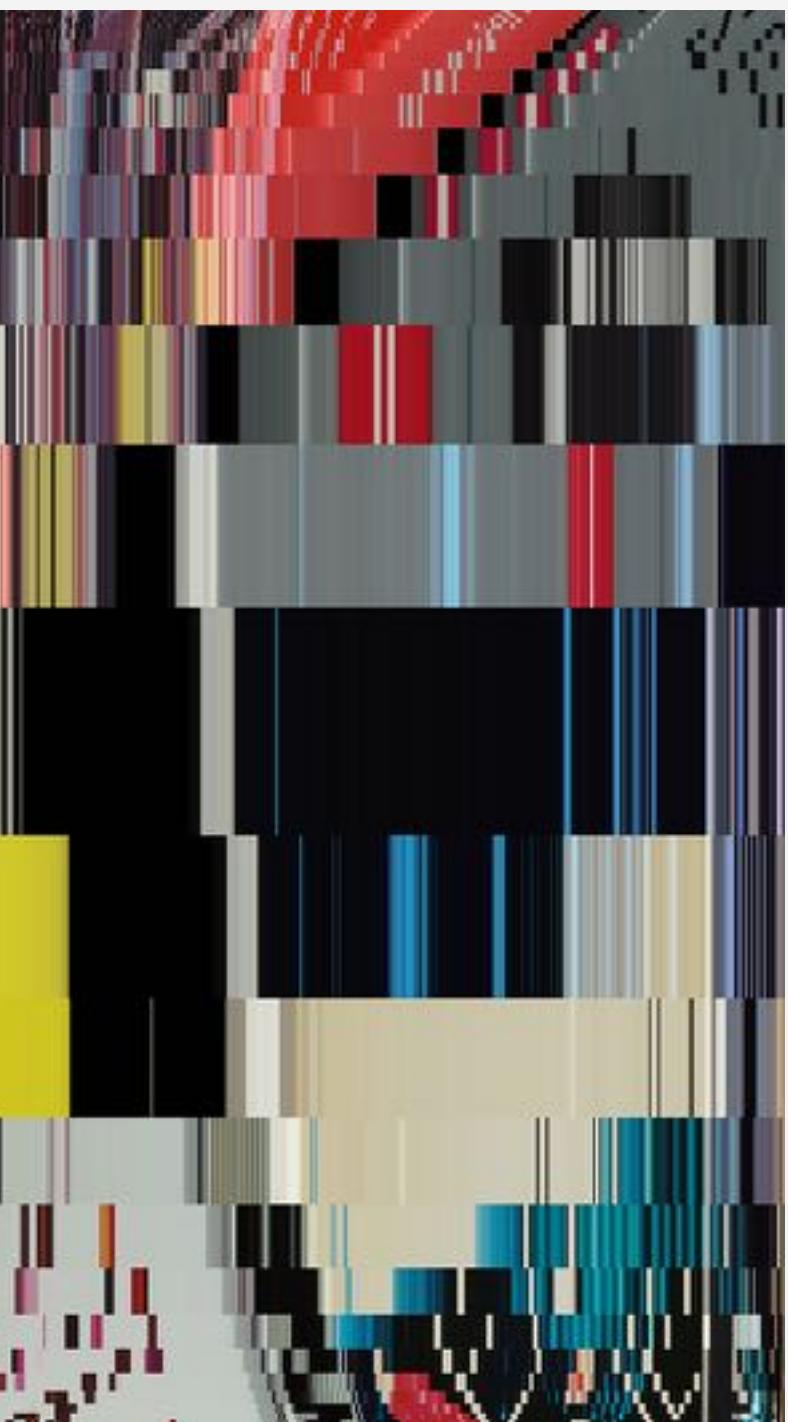
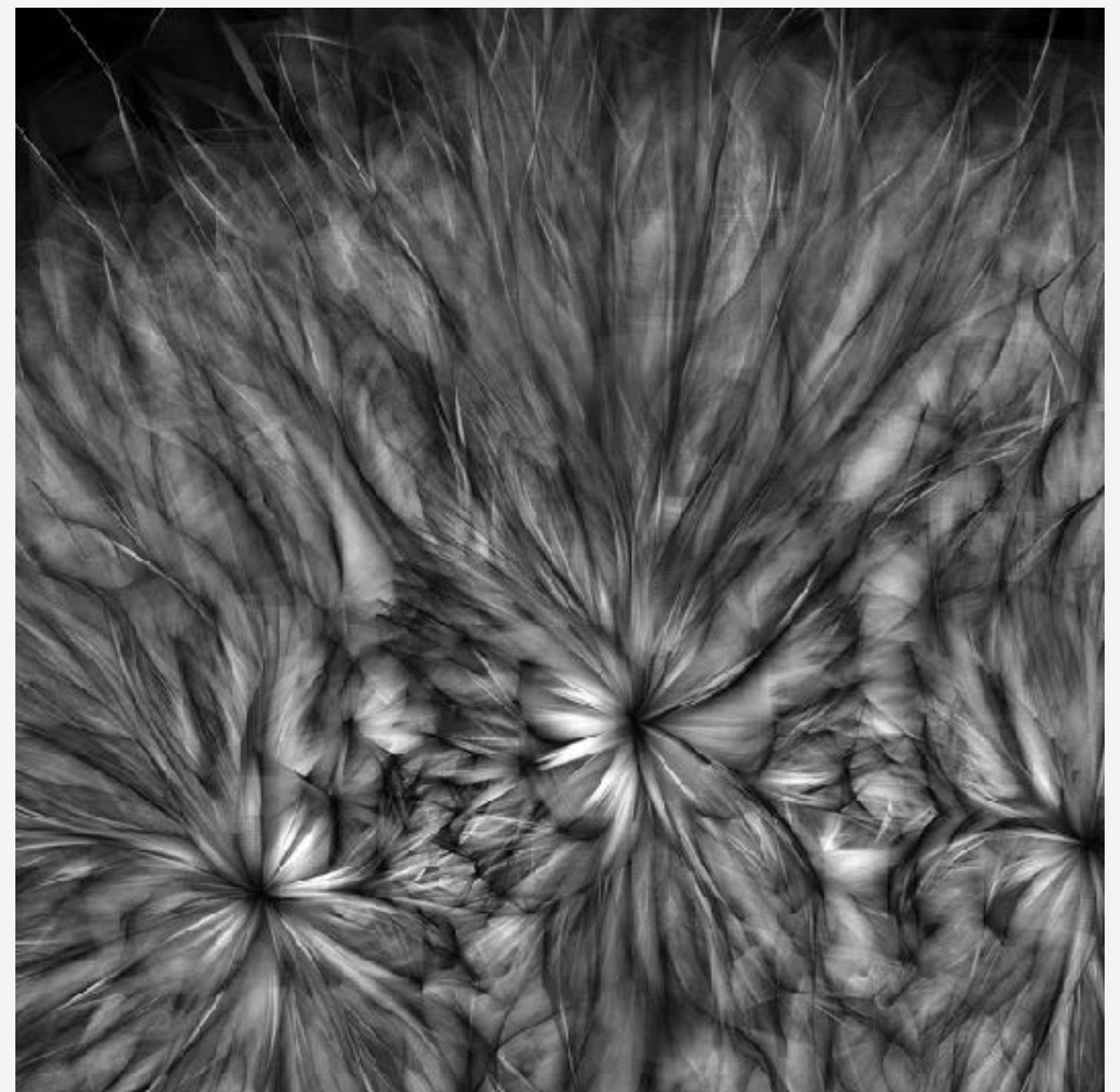
Herbert W. Franke



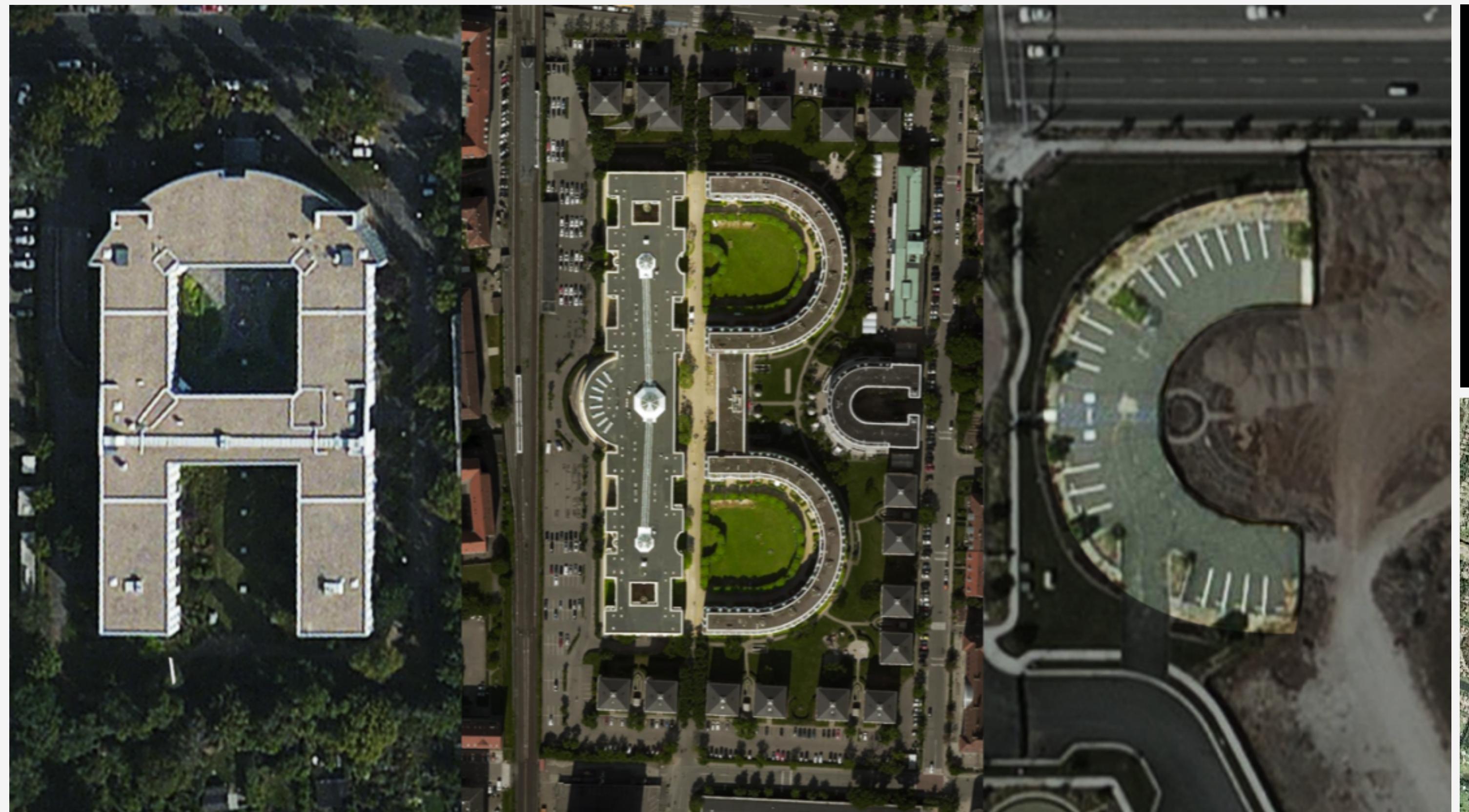
Manfred Mohr



Andreas Hoff



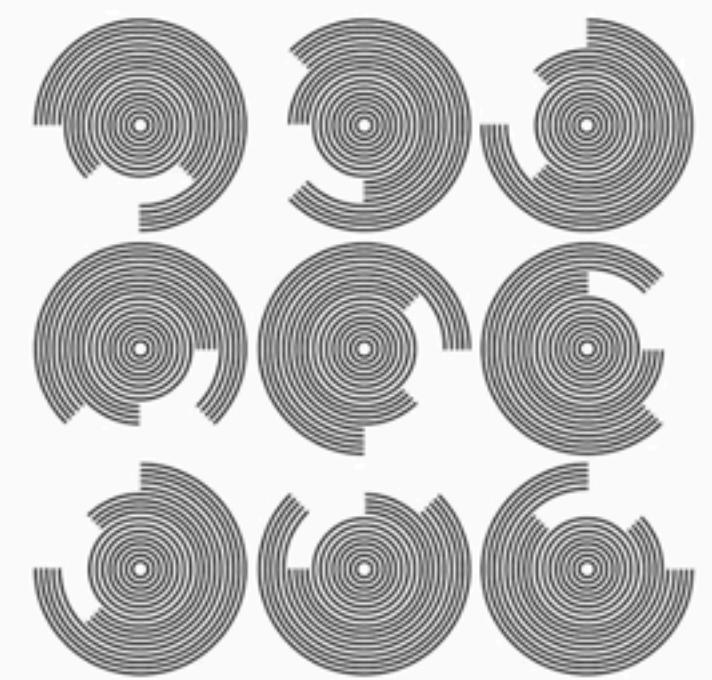
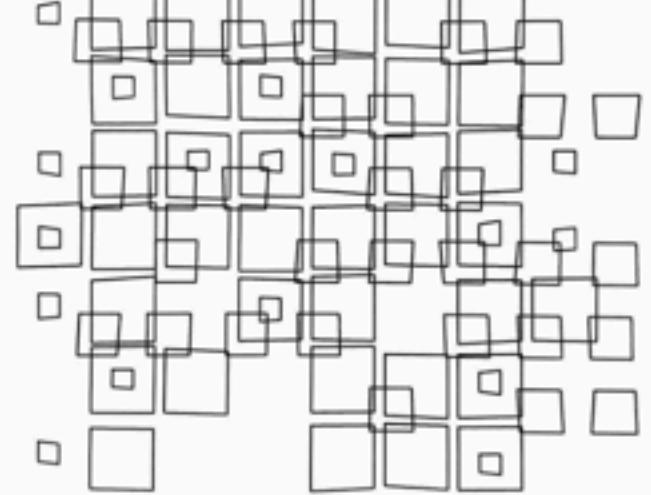
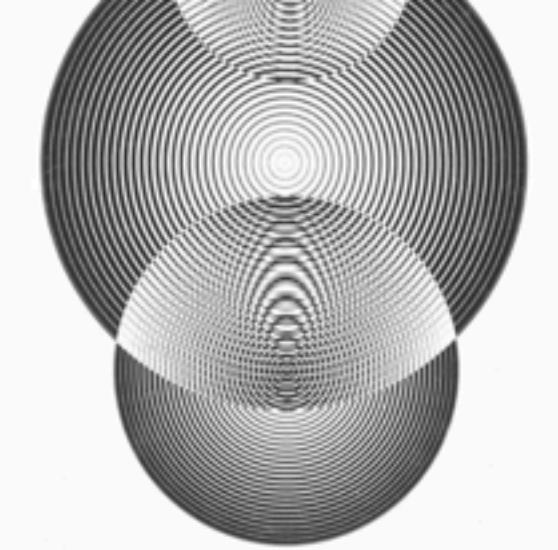
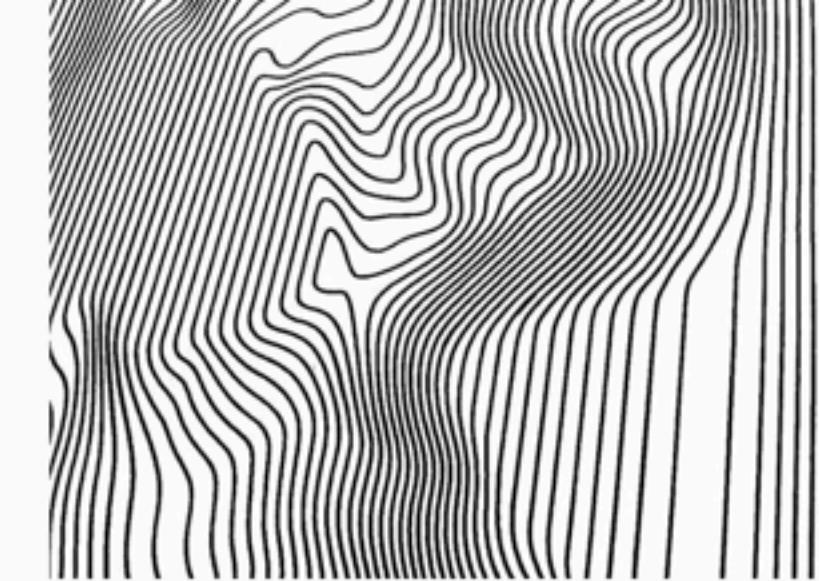
Casey Reas



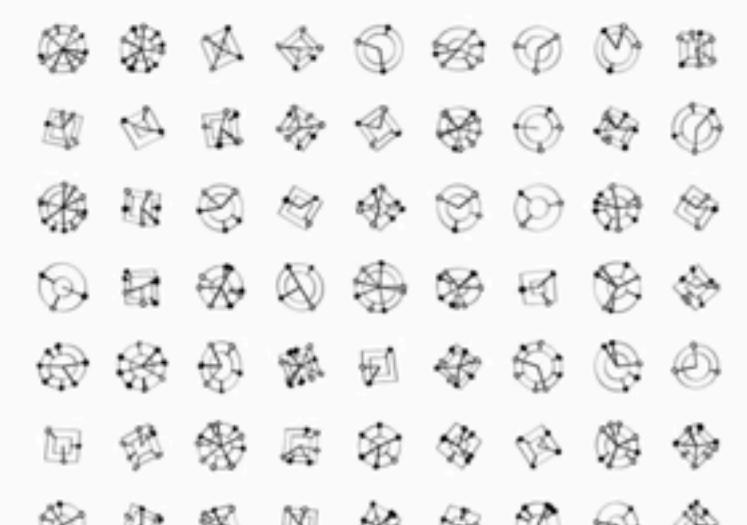
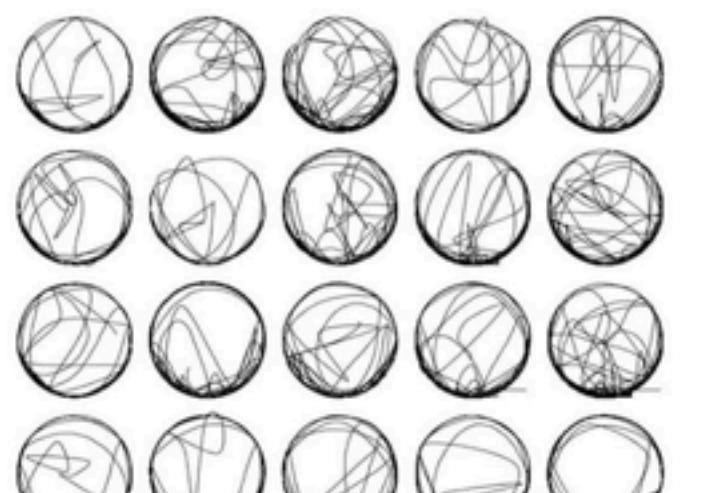
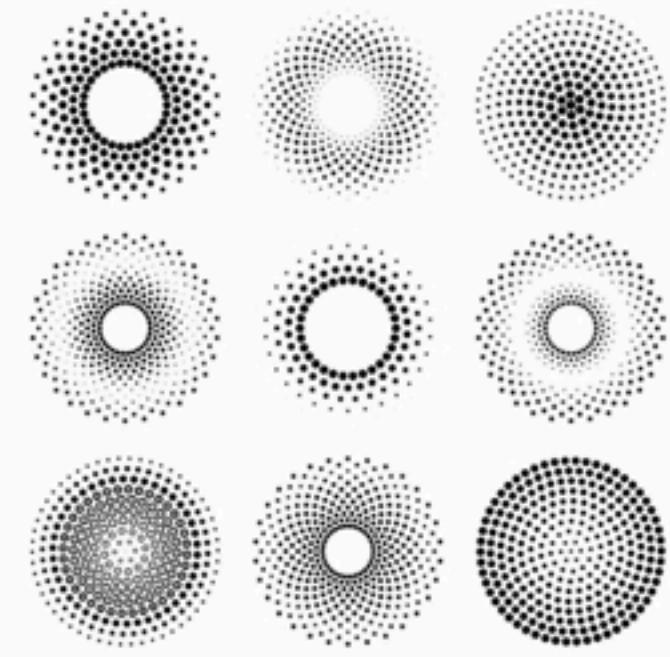
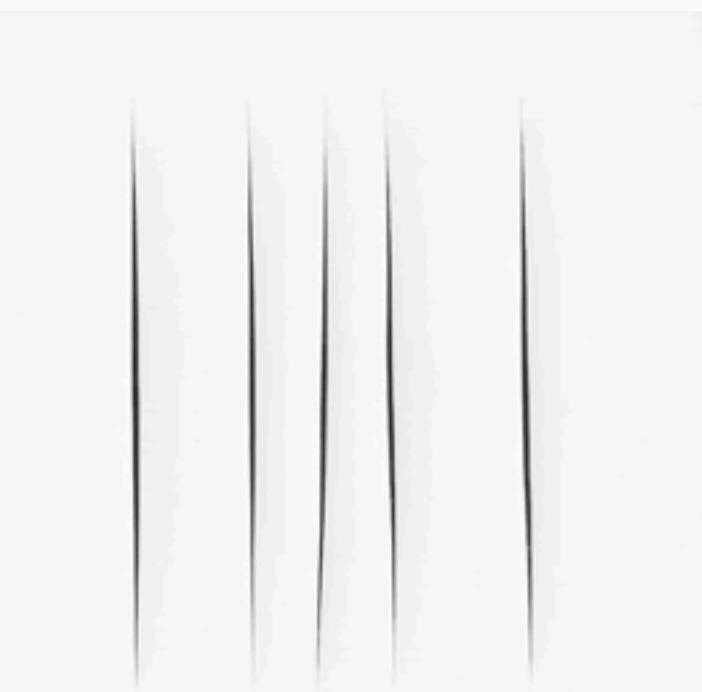
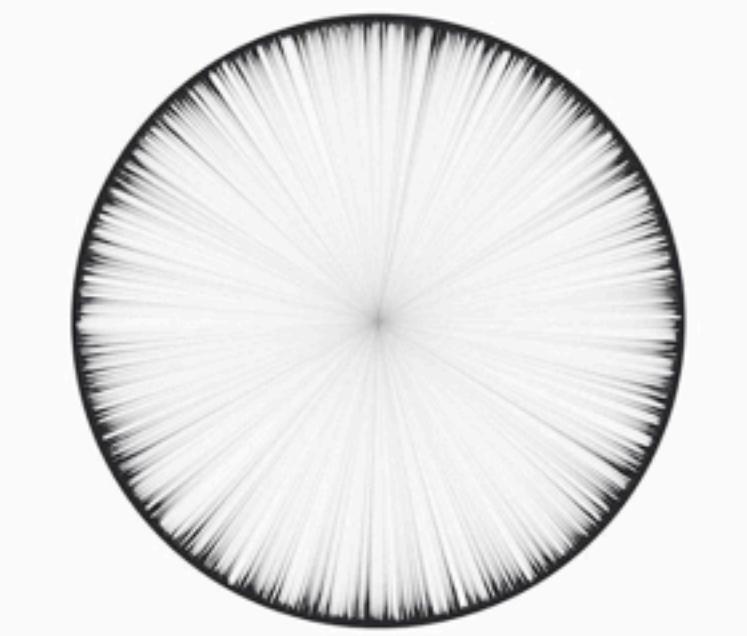
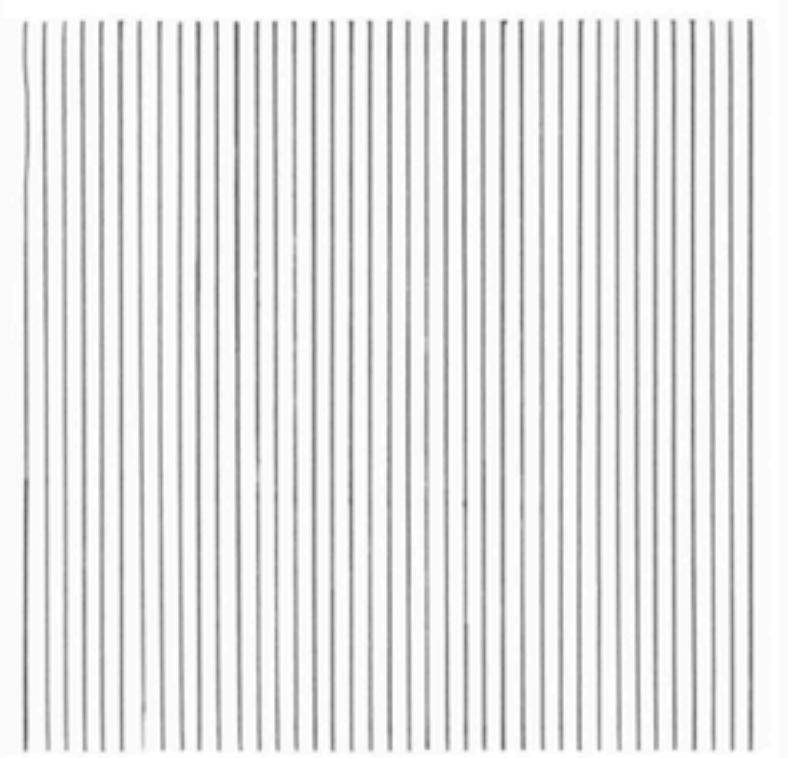
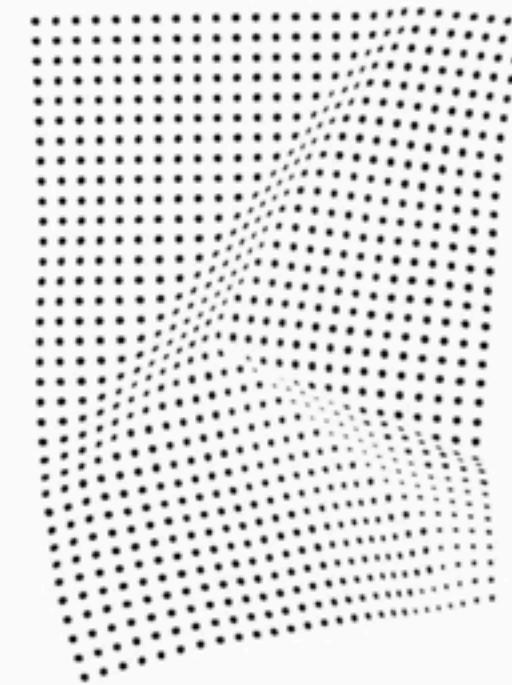
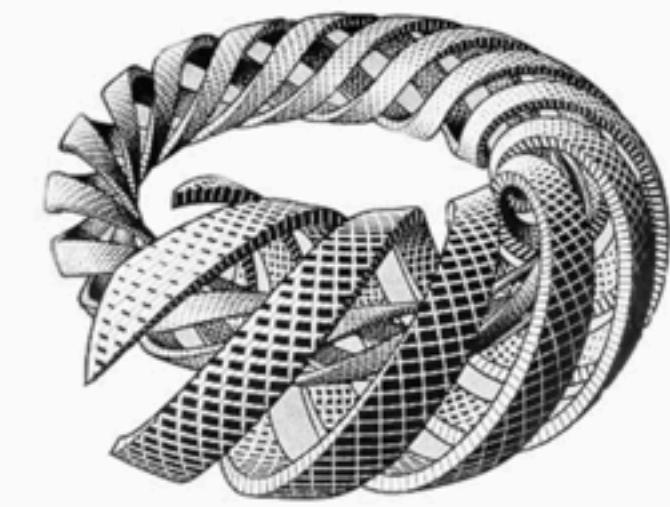
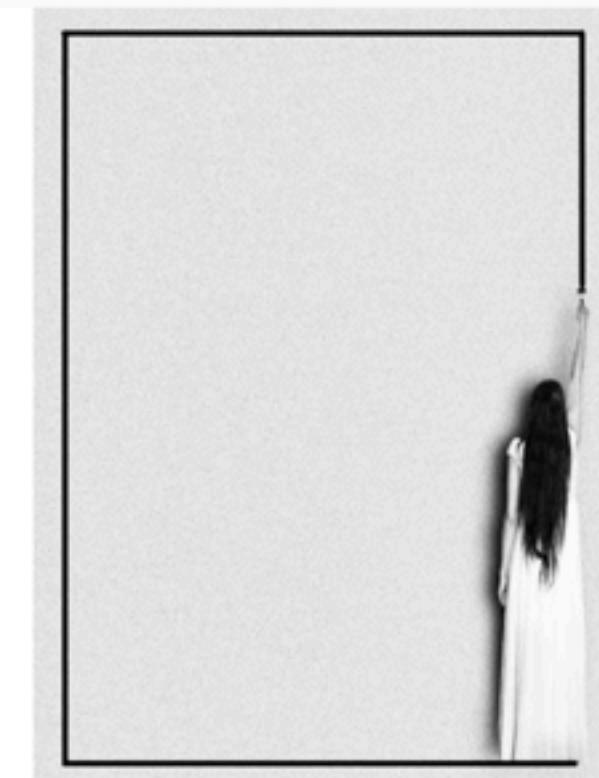
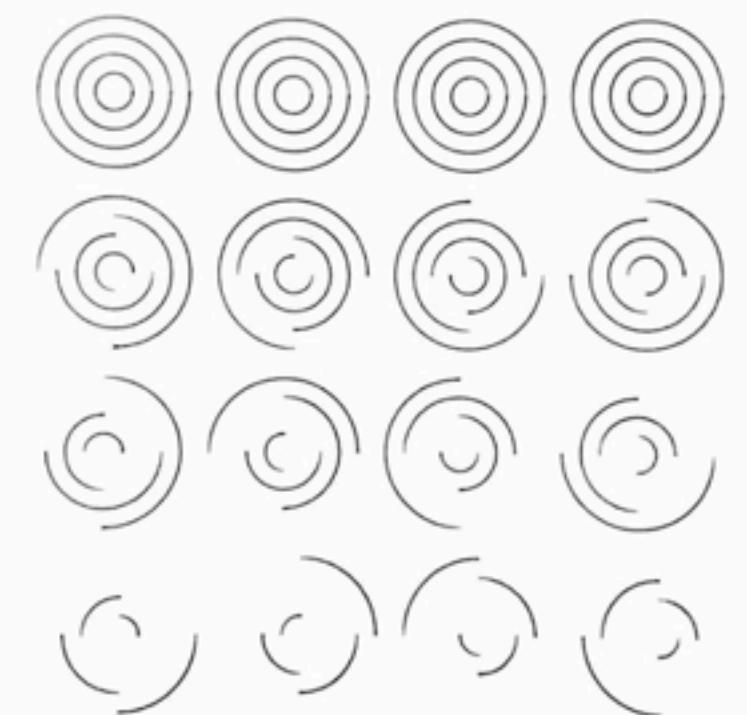
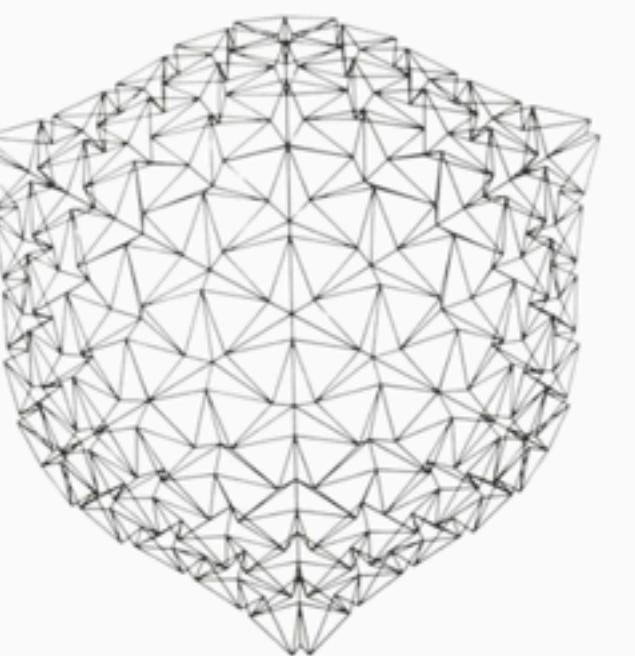
KICKSTART THE
PLATETRY SEARCH
FOR LETTERFORMS



Benedikt Groß



flowerflowerflowerflower
flowerflowerflowerflower



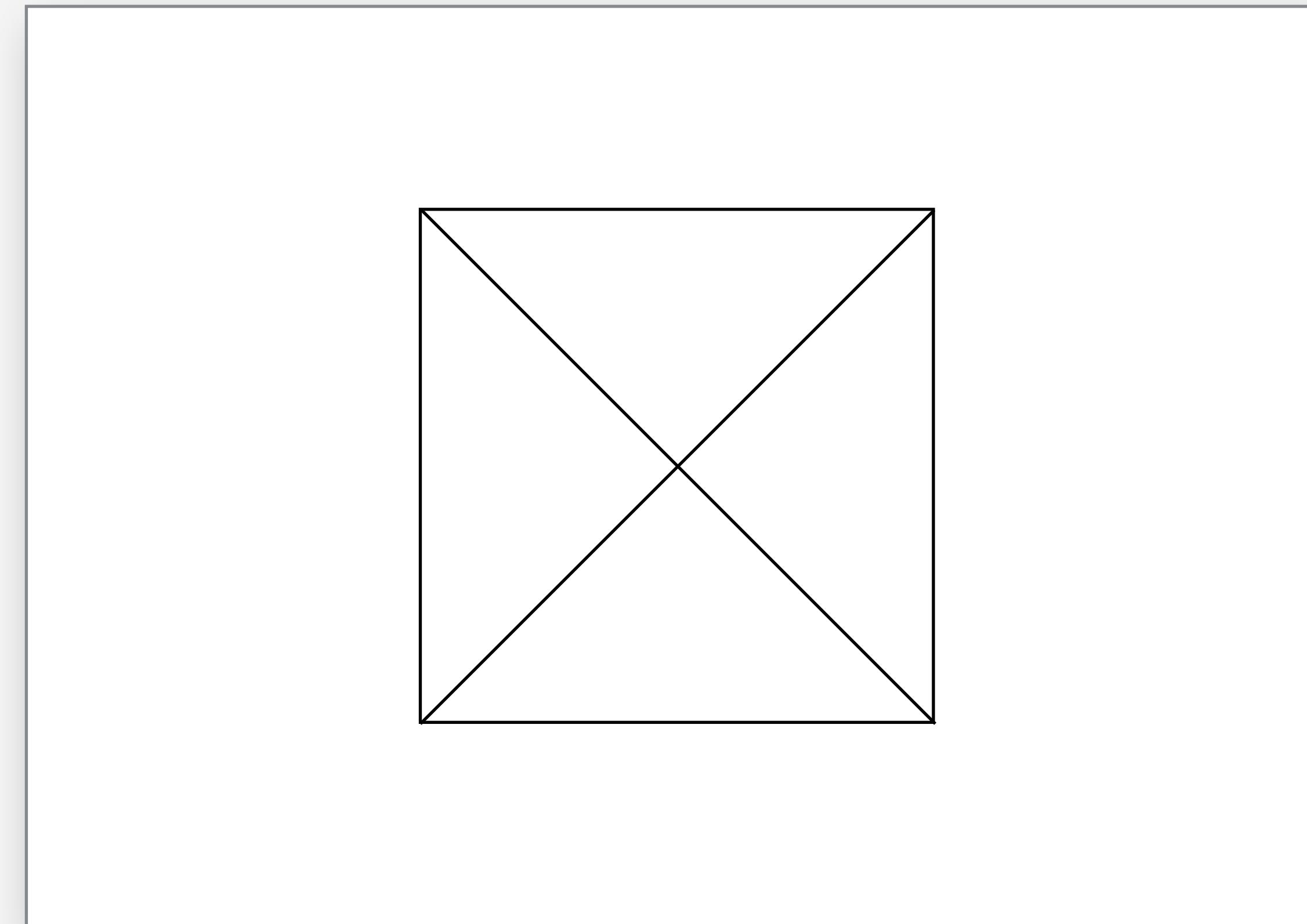
<https://www.instagram.com/in.white.rooms/?hl=en>

PROCEDURAL DRAWING LANGUAGE

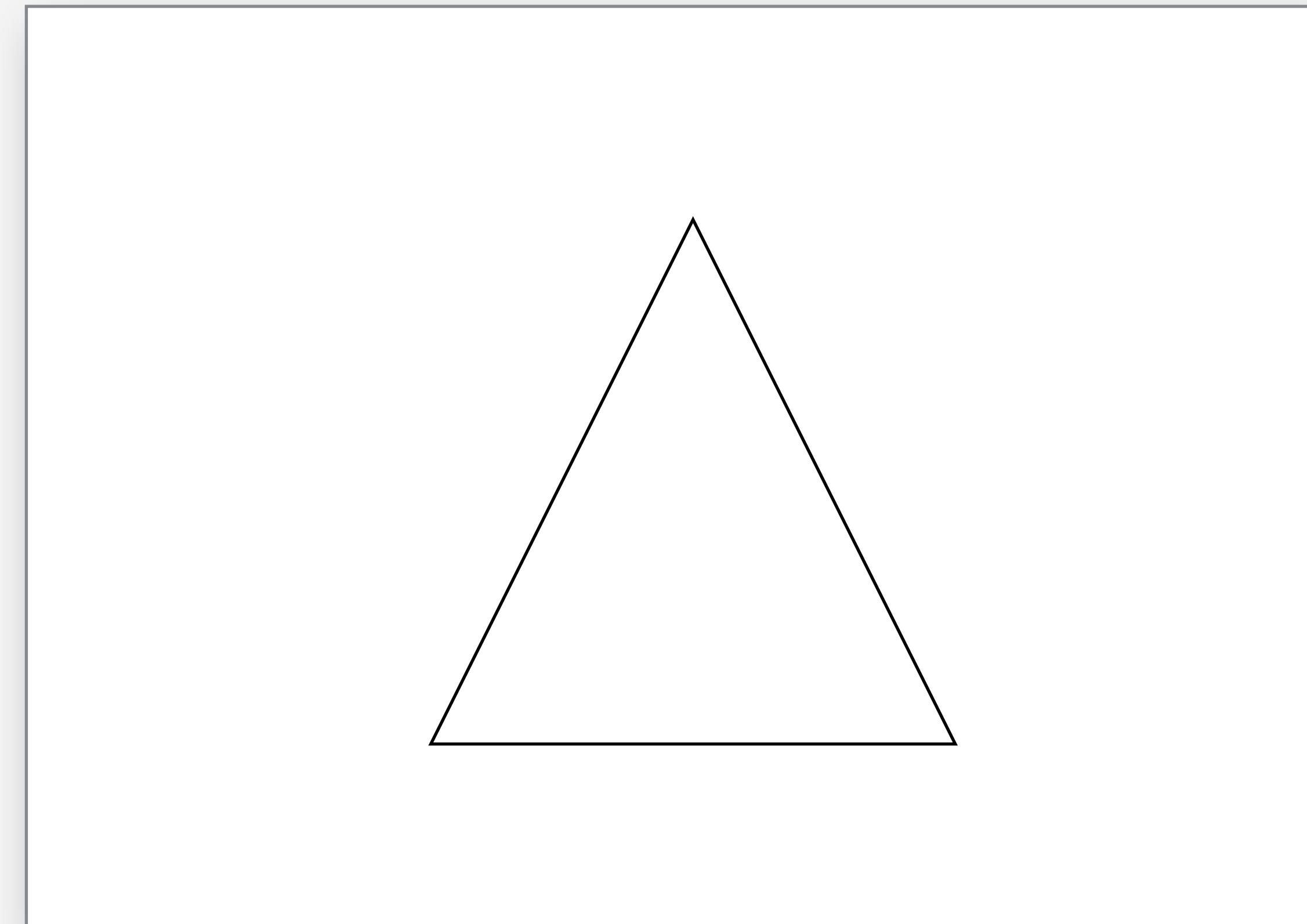
Class exercise

1. Partner up.
2. Get a pen and some papers
3. The person drawing is blindfolded
4. Agree on a set of rules for drawing the following images

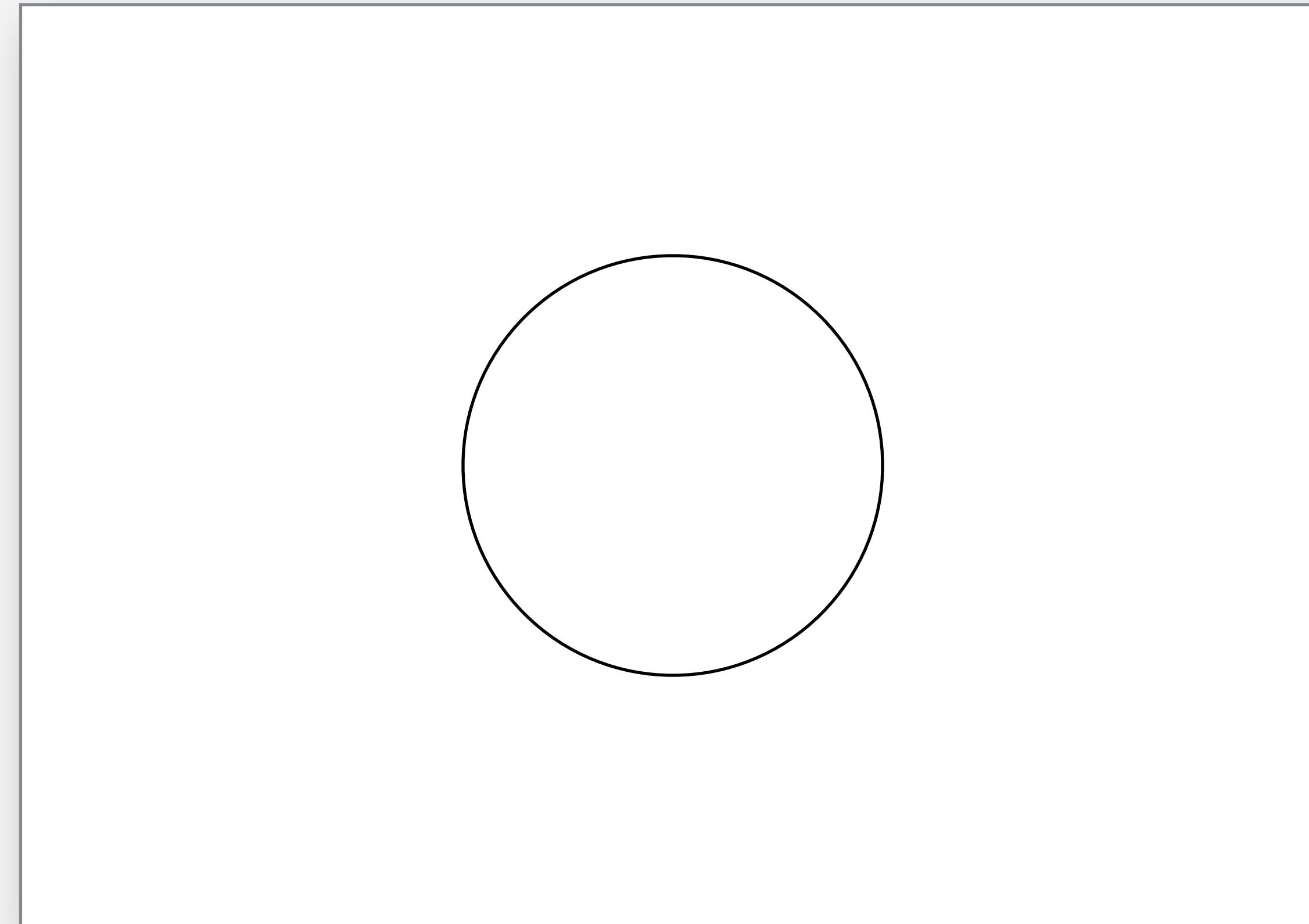
DRAWING LANGUAGE



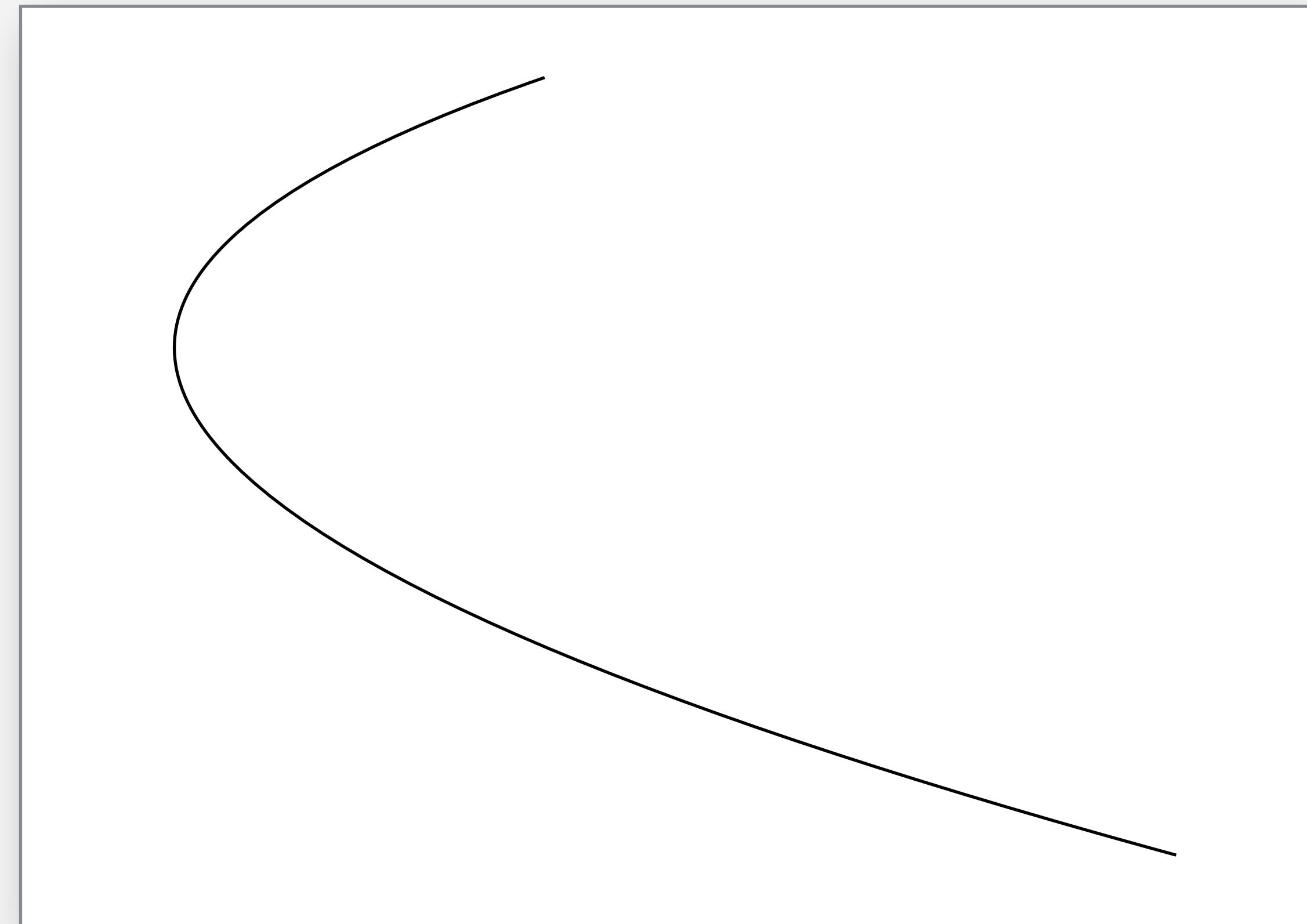
DRAWING LANGUAGE



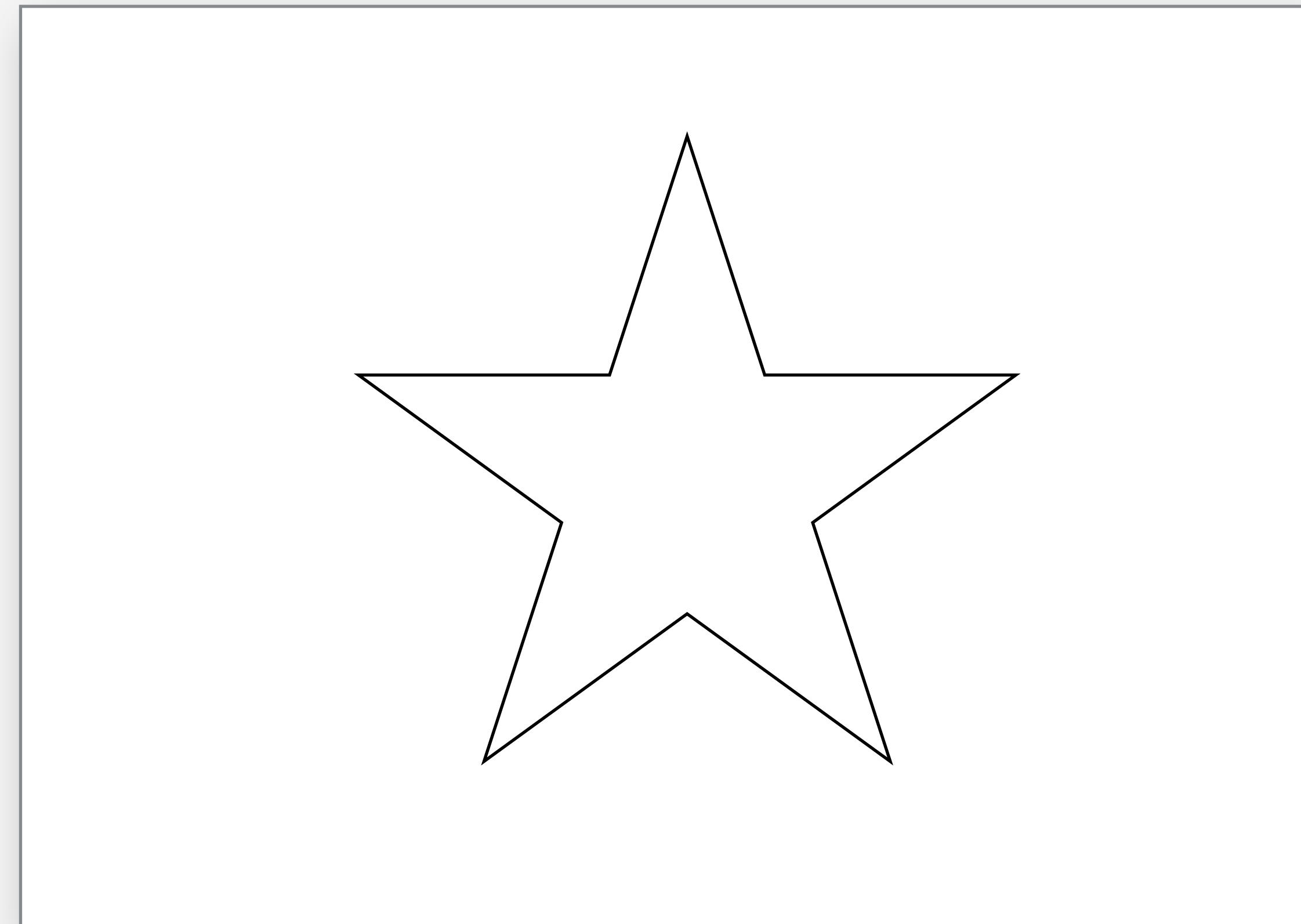
DRAWING LANGUAGE



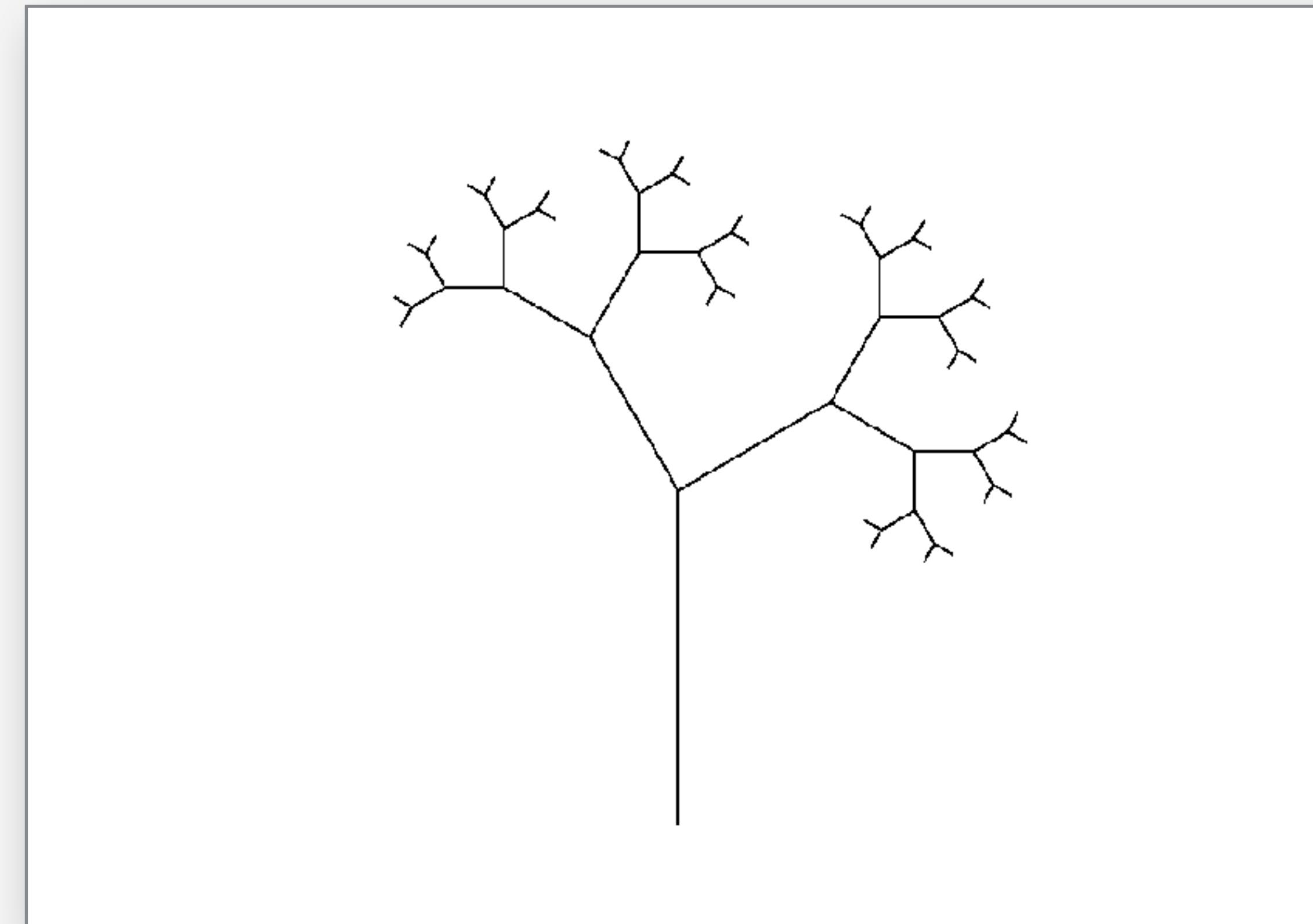
DRAWING LANGUAGE



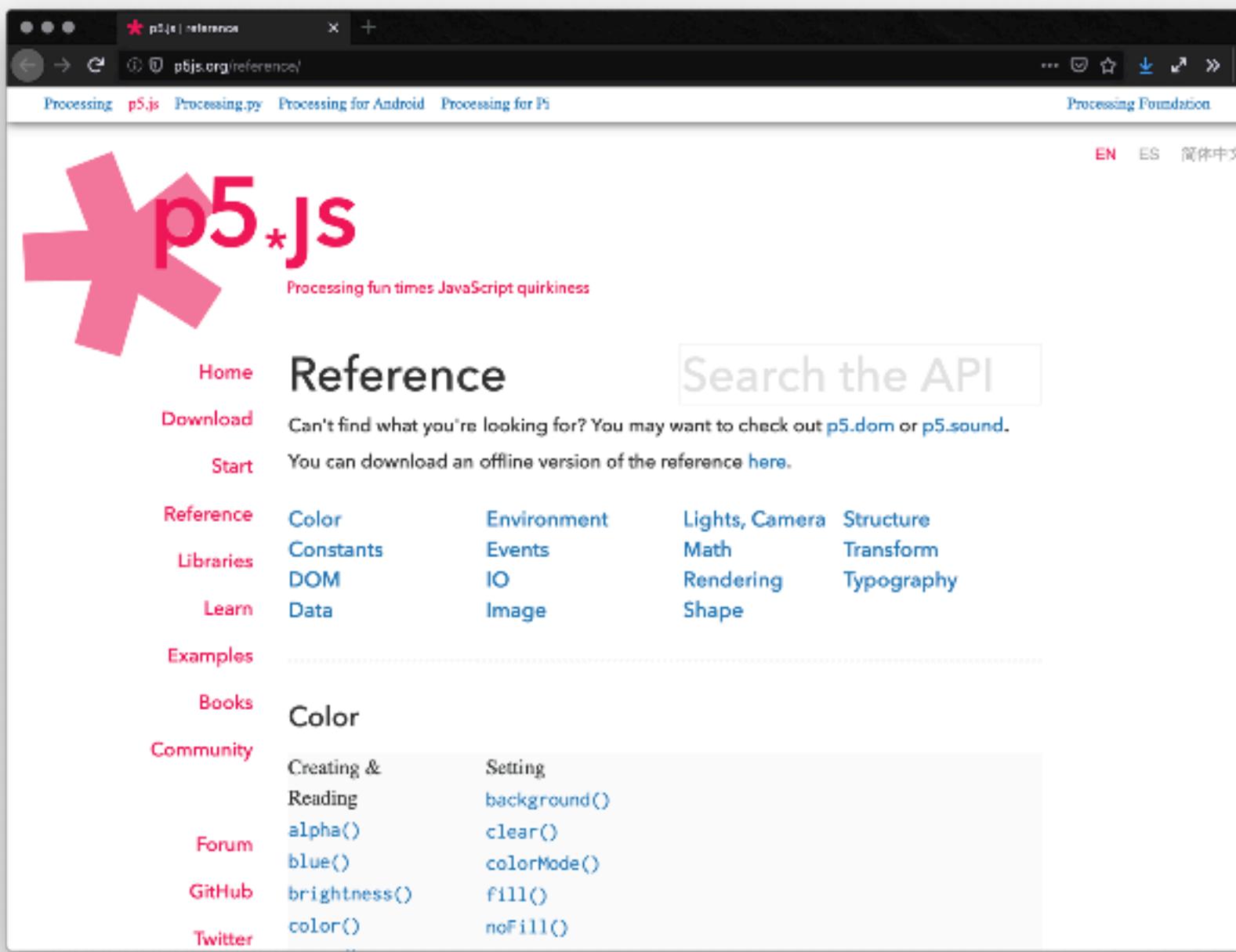
DRAWING LANGUAGE



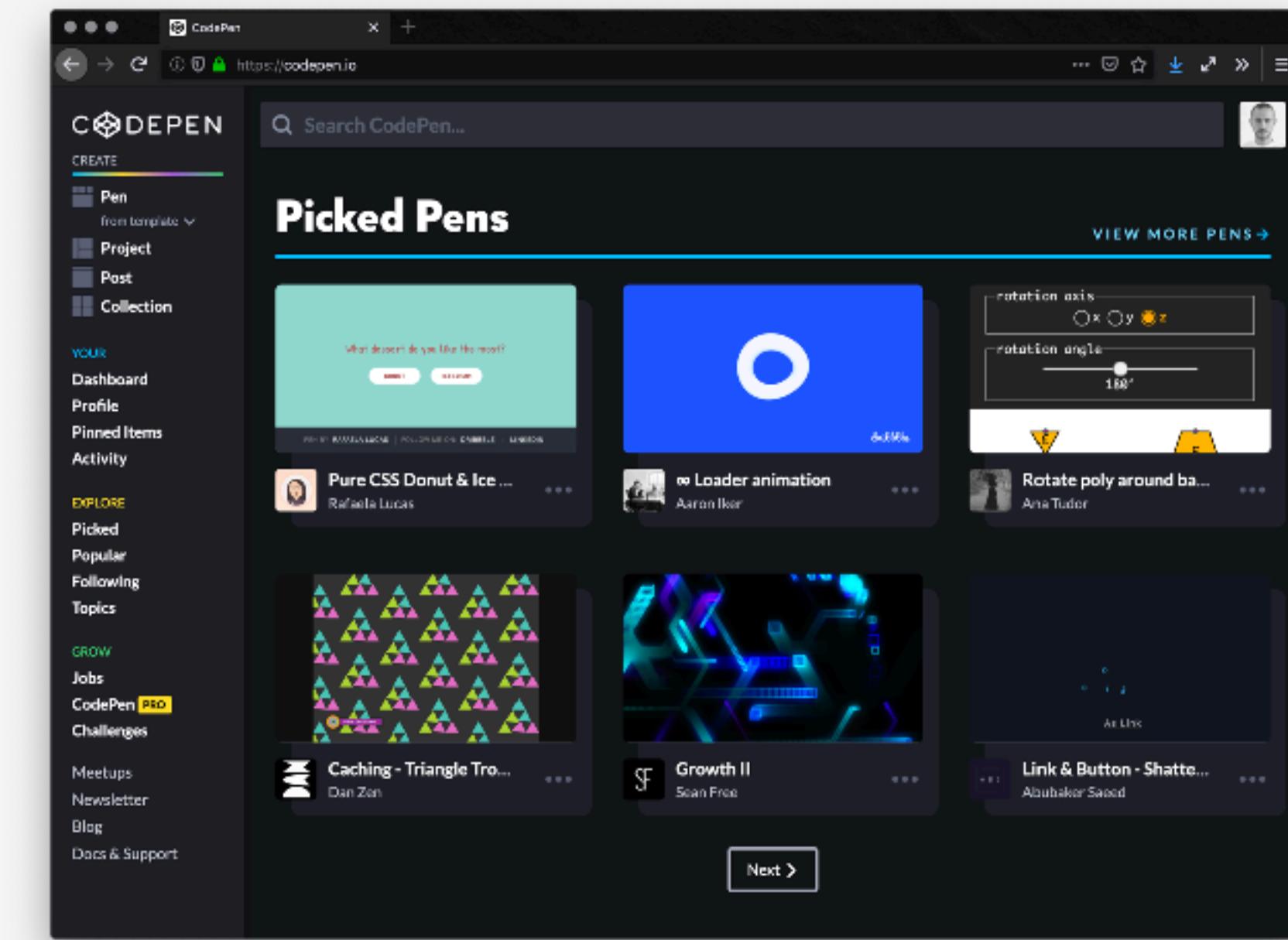
DRAWING LANGUAGE



OUR WORKSPACE:



p5js.org/reference



codepen.io

2D DRAWING COMMANDS

`size(width, height)`

`background(r, g, b)`

`stroke(r, g, b, a)`

`fill(r, g, b, a)`

`strokeWeight(w)`

`noStroke()`

`noFill()`

`smooth(level)`

`save("imgName.png")`

`point(x, y)`

`line(x1, y1, x2, y2)`

`rect(x, y, width, height)`

`ellipse(x, y, width, height)`

`arc(x, y, width, height, start, stop)`

`quad(x1, y1, x2, y2, x3, y3, x4, y4)`

`triangle(x1, y1, x2, y2, x3, y3)`

`curve(x1, y1, x2, y2, x3, y3, x4, y4)`

`bezier(x1, y1, x2, y2, x3, y3, x4, y4)`

CLASS EXERCISE

Pick any object in the room and create an abstract representation of it
using the p5 drawing commands.

HOMEWORK

1. Self portrait

Please program a self portrait in p5 using any commands and in any visual style you like.

Be mindful of details and make purposeful visual decisions.

* Submission via Slack, ie: 10:03 lior ben gai Week 1 : <https://codepen.io/soogbet/pen/MMMjZN>

2. Watch:

Hello World! Processing (film) - <https://vimeo.com/60735314> [40 minutes]