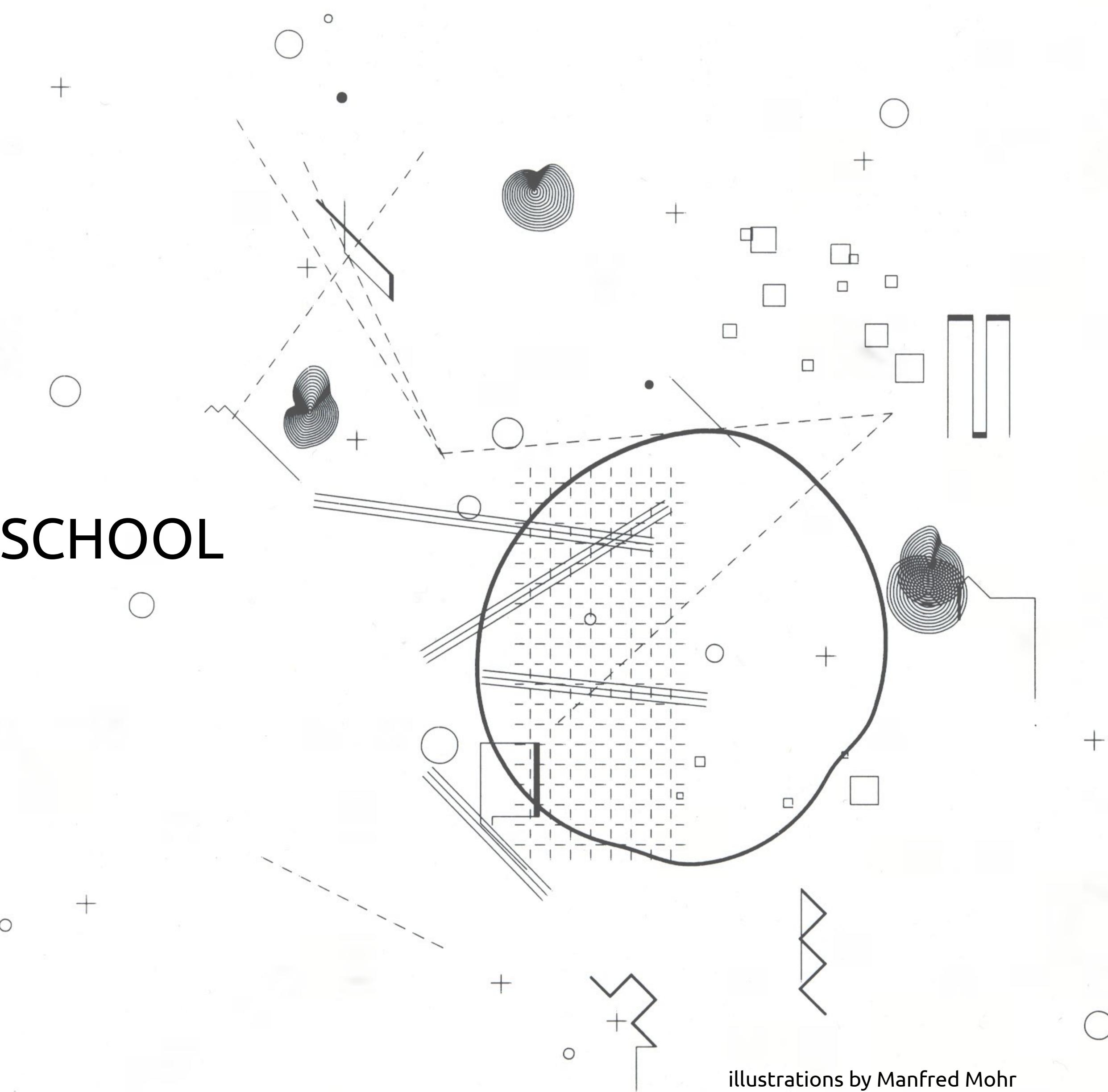


# CREATIVE COMPUTING 2019 SUMMER SCHOOL

Introduction to graphics coding

August 2019

**Lior Ben-Gai**



illustrations by Manfred Mohr

# GOALS

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

## PROGRAMMING SKILLS

1. **Beginner** (~ 25%) - Little to no experience in computer programming and/or graphics
2. **Intermediate** (~ 65%) - Basic programming and familiarity with computer graphics
3. **Pro** (~ 10%) - Full proficiency in programming computer graphics, animation and human interaction.

## GENERAL TOPICS:

1. Introduction to programming and procedural drawing
2. Variables, Functions, Conditions and Loops
3. Basic interaction: (Mouse / Keyboard )
4. Data structures (Array, Object)
5. Transformations and number generators
6. Sound and data Visualization

# SCHEDULE FOR THIS WEEK

9:00 - 12:30

Graphics workshop (Group1) LIOR VALERIO ALEX	Graphics workshop (Group1) LIOR VALERIO ALEX	Graphics workshop (Group1) LIOR VALERIO ALEX	Graphics workshop (Group1) LIOR VALERIO ALEX
--	--	--	--

13:30 - 16:00

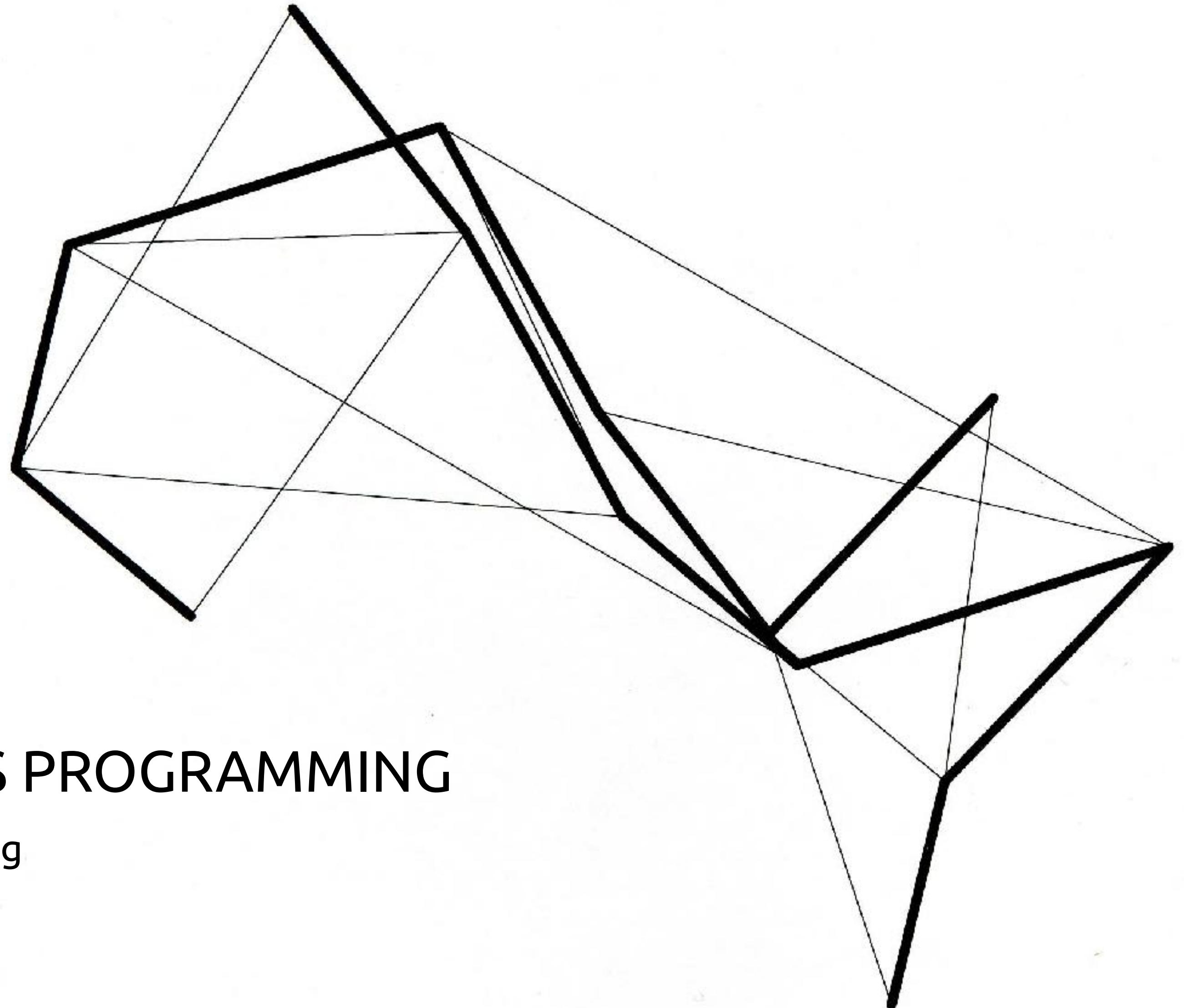
Tutorial Study (2.5 HOURS)  LIOR MEGAN ROB VALERIO ALEX	Tutorial Study (2.5 HOURS)  LIOR and STEPHANIE MEGAN ROB VALERIO ALEX	Presentations 1pm -3pm (2 HOURS)  LIOR ALICE VALERIO ALEX MEGAN ROB
--	--	--

# SESSIONS (4 OUT OF 6)

	<b>title</b>	<b>topics</b>	<b>study</b>
<b>1</b>	introduction	procedural drawing, visual expression	self portrait
<b>2</b>	building clocks	variables, conditions, functions, loops	clocks
<b>3</b>	generators	random, noise, sine	generative drawing app
<b>4</b>	data visualization	external data, objects, arrays	world views
<b>5</b>	visuals of earth	FFT analysis, sound visualization	VOE
<b>6</b>	digital audio synthesis	digital sound, interactive apps	sound toy

## COURSE INDEX

<https://soogbet.github.io/goldsmiths-summer19/>



# INTRODUCTION TO GRAPHICS PROGRAMMING

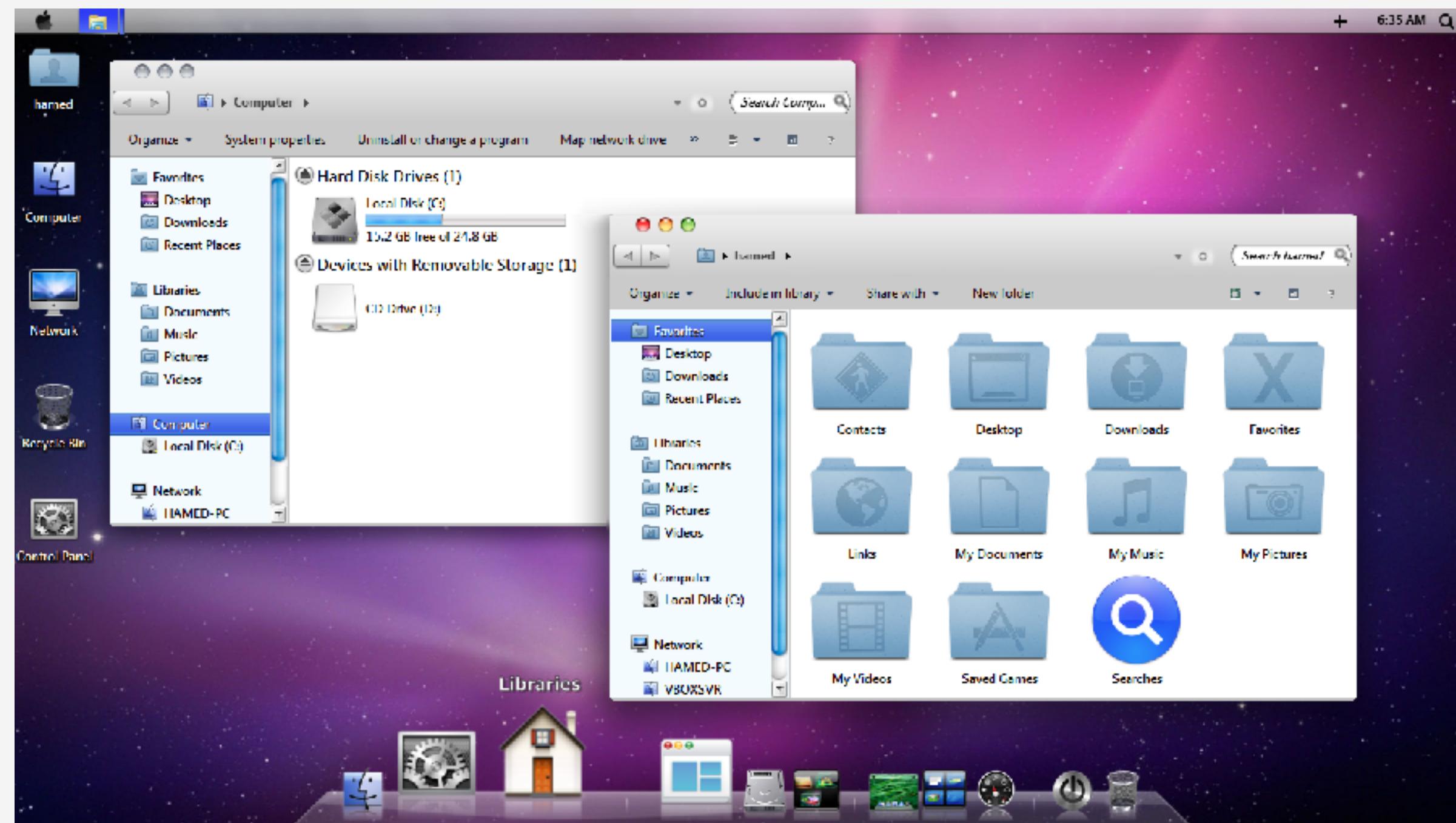
Programming / Language / Procedural drawing

WHAT IS COMPUTER PROGRAMMING ANYWAY?

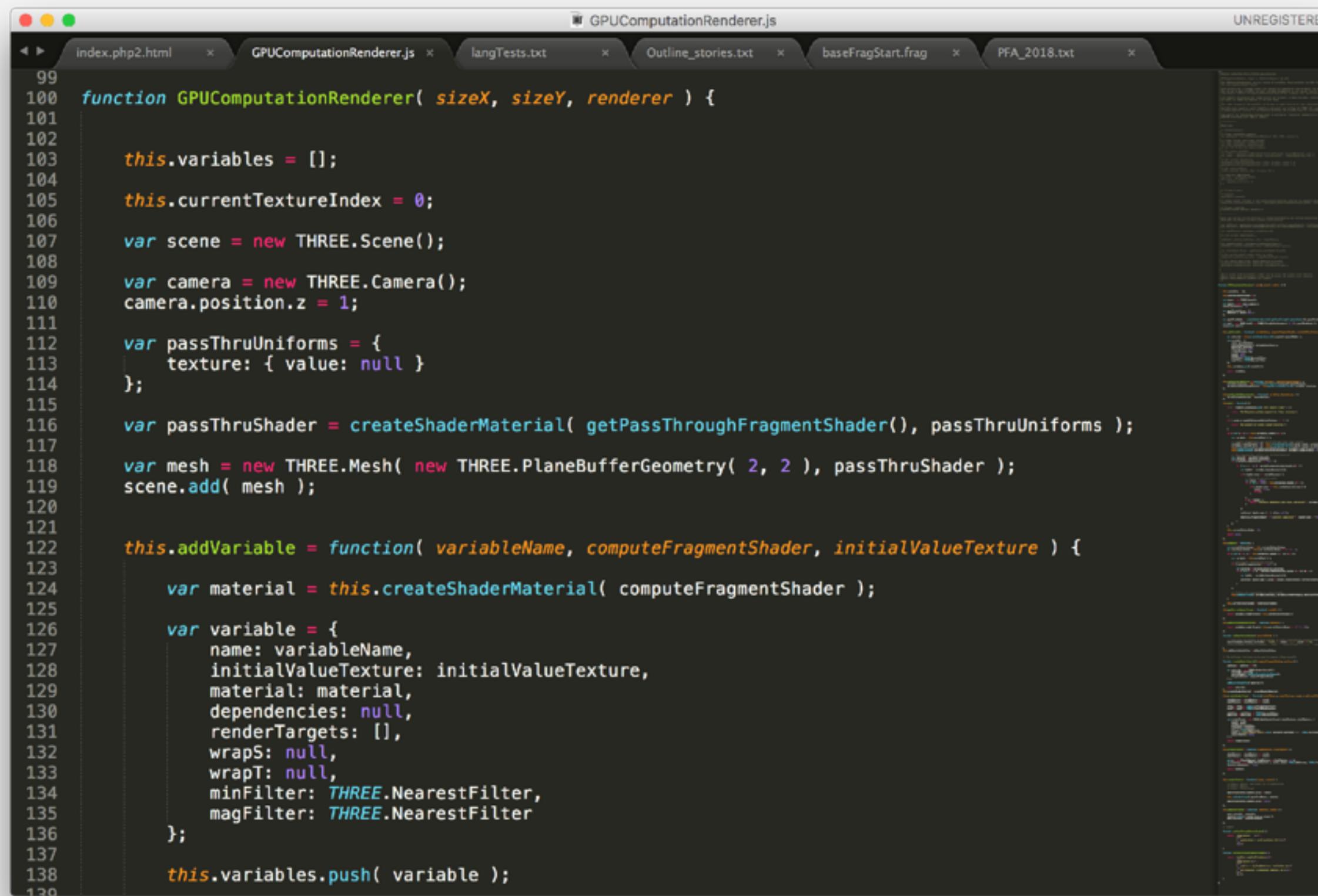
# IS THIS COMPUTER PROGRAMMING?



# IS THIS COMPUTER PROGRAMMING?



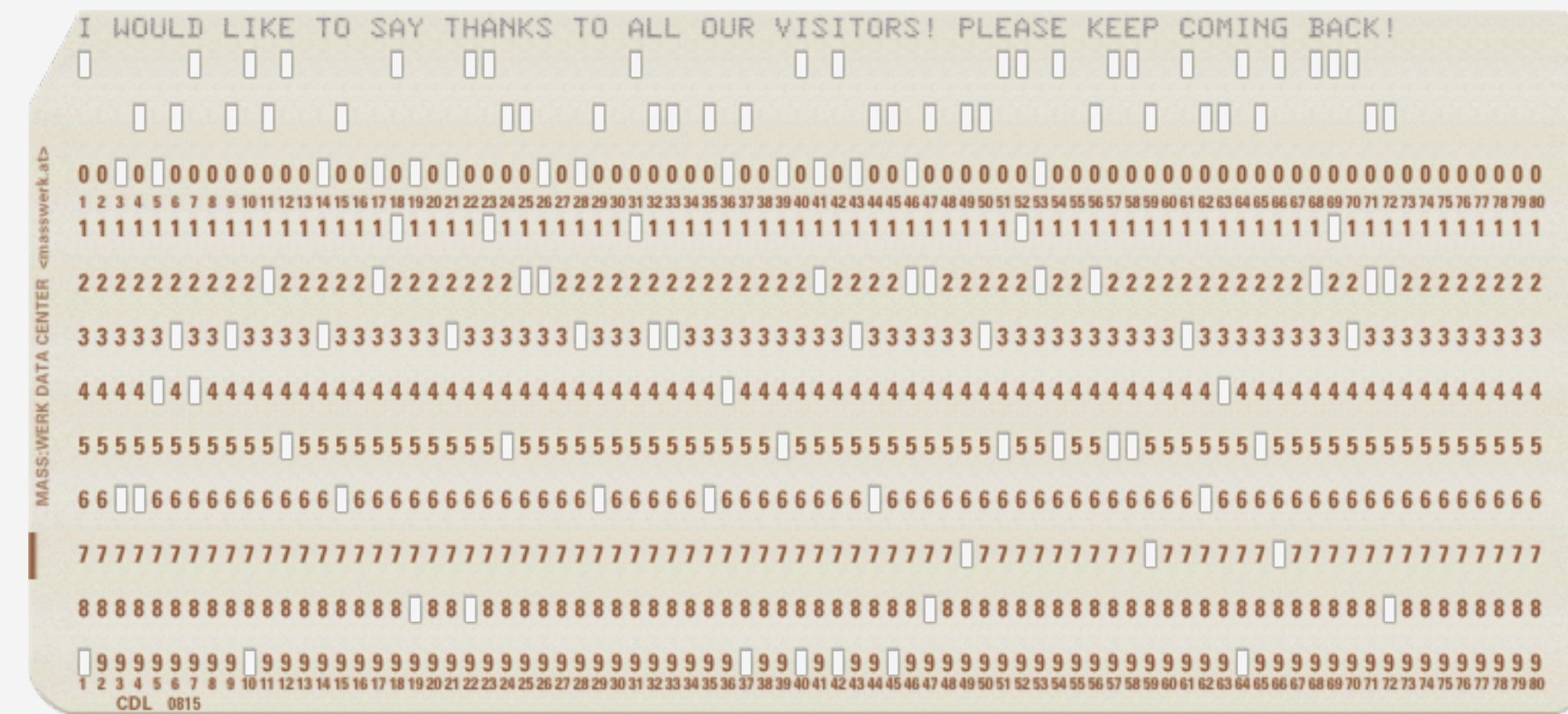
# IS THIS COMPUTER PROGRAMMING?



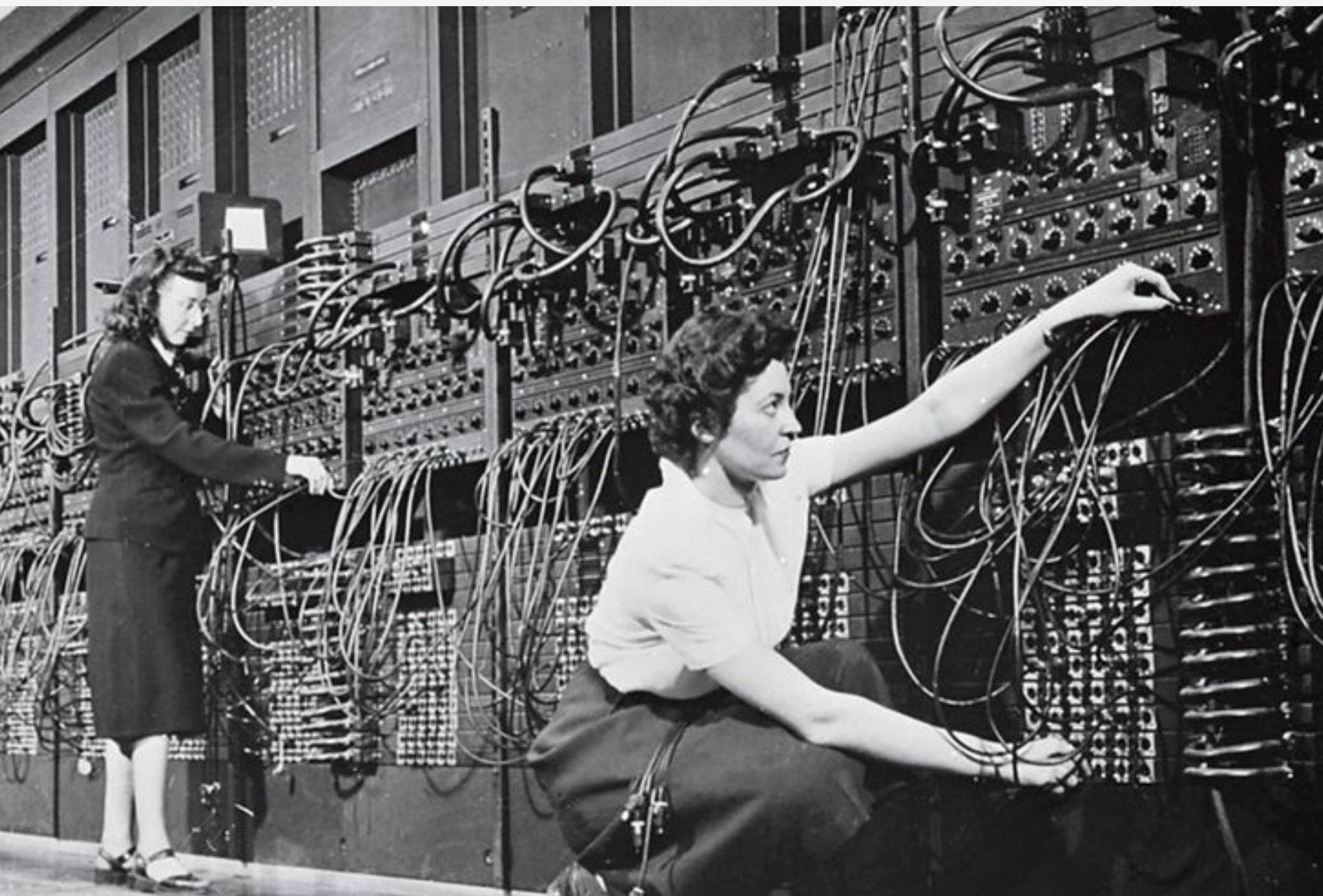
A screenshot of a code editor window titled "GPUComputationRenderer.js". The window shows several tabs at the top: "index.php2.html", "GPUComputationRenderer.js", "langTests.txt", "Outline\_stories.txt", "baseFragStart.frag", and "PFA\_2018.txt". The main pane displays a block of JavaScript code with line numbers from 99 to 138. The code defines a constructor for a GPUComputationRenderer class, initializes scene, camera, and material components, and implements an addVariable method to manage shader variables.

```
99 function GPUComputationRenderer( sizeX, sizeY, renderer ) {
100
101     this.variables = [];
102
103     this.currentTextureIndex = 0;
104
105     var scene = new THREE.Scene();
106
107     var camera = new THREE.Camera();
108     camera.position.z = 1;
109
110     var passThruUniforms = {
111         texture: { value: null }
112     };
113
114     var passThruShader = createShaderMaterial( getPassThroughFragmentShader(), passThruUniforms );
115
116     var mesh = new THREE.Mesh( new THREE.PlaneBufferGeometry( 2, 2 ), passThruShader );
117     scene.add( mesh );
118
119
120
121     this.addVariable = function( variableName, computeFragmentShader, initialValueTexture ) {
122
123         var material = this.createShaderMaterial( computeFragmentShader );
124
125         var variable = {
126             name: variableName,
127             initialValueTexture: initialValueTexture,
128             material: material,
129             dependencies: null,
130             renderTargets: [],
131             wrapS: null,
132             wrapT: null,
133             minFilter: THREE.NearestFilter,
134             magFilter: THREE.NearestFilter
135         };
136
137         this.variables.push( variable );
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
```

# IS THIS COMPUTER PROGRAMMING?



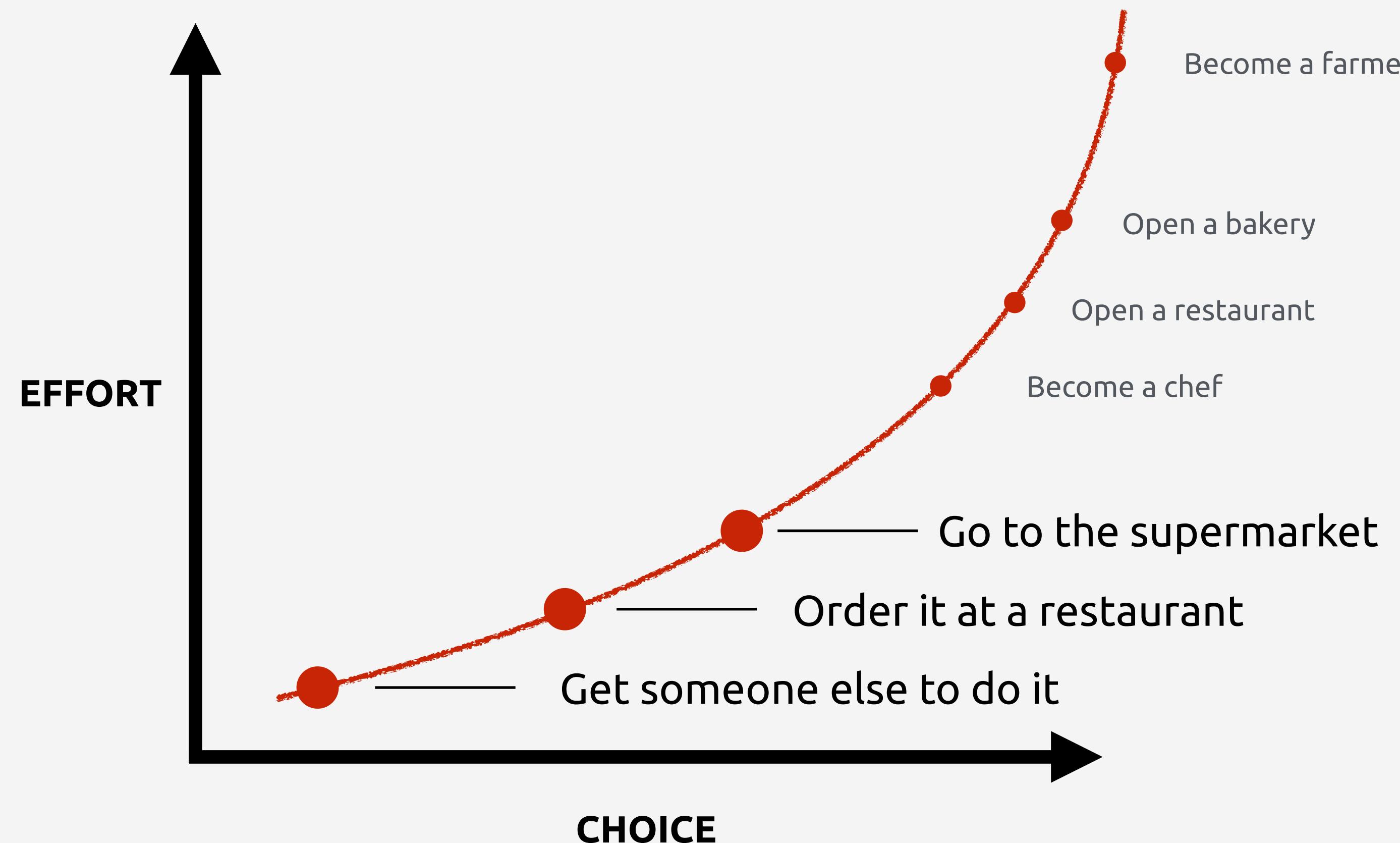
# IS THIS COMPUTER PROGRAMMING?



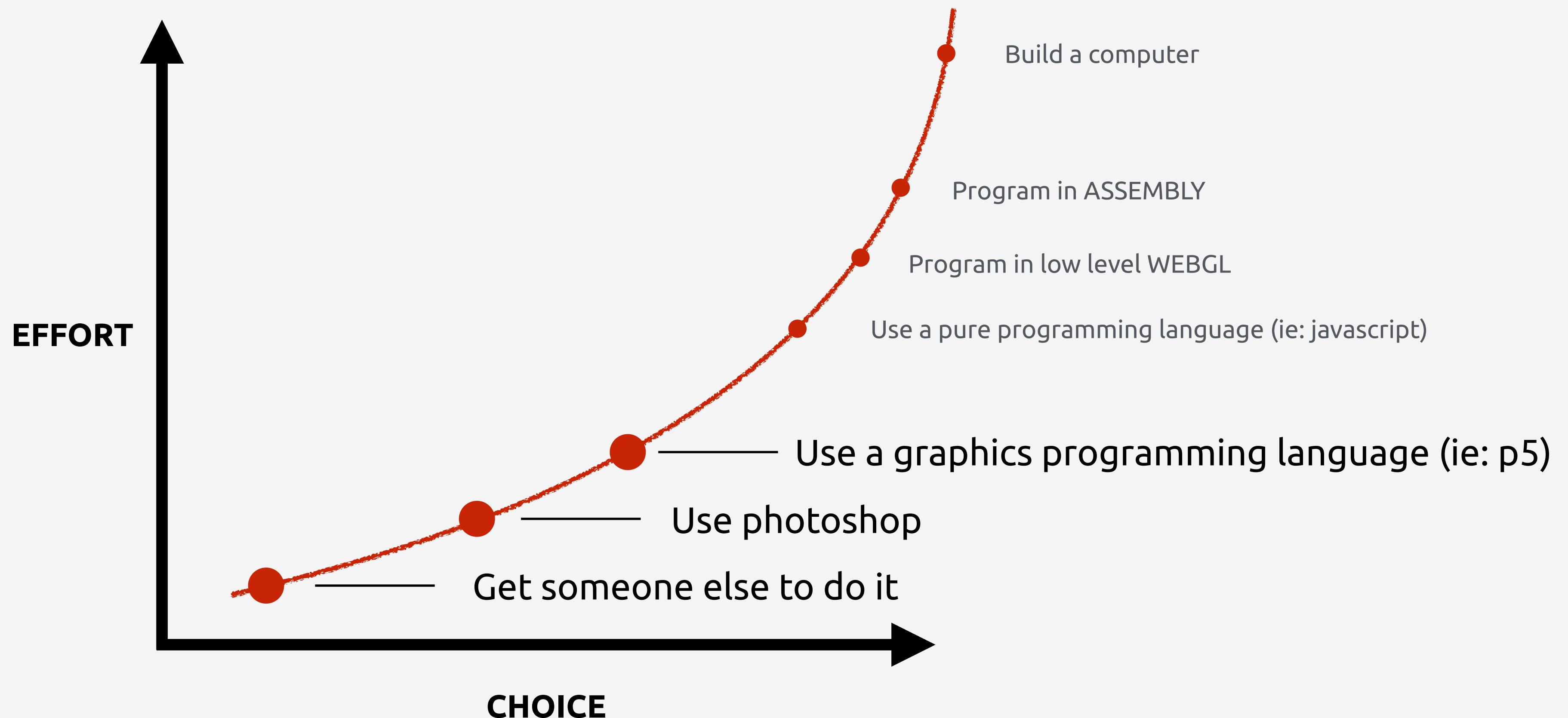
# IS THIS COMPUTER PROGRAMMING?



# MAKING A SANDWICH

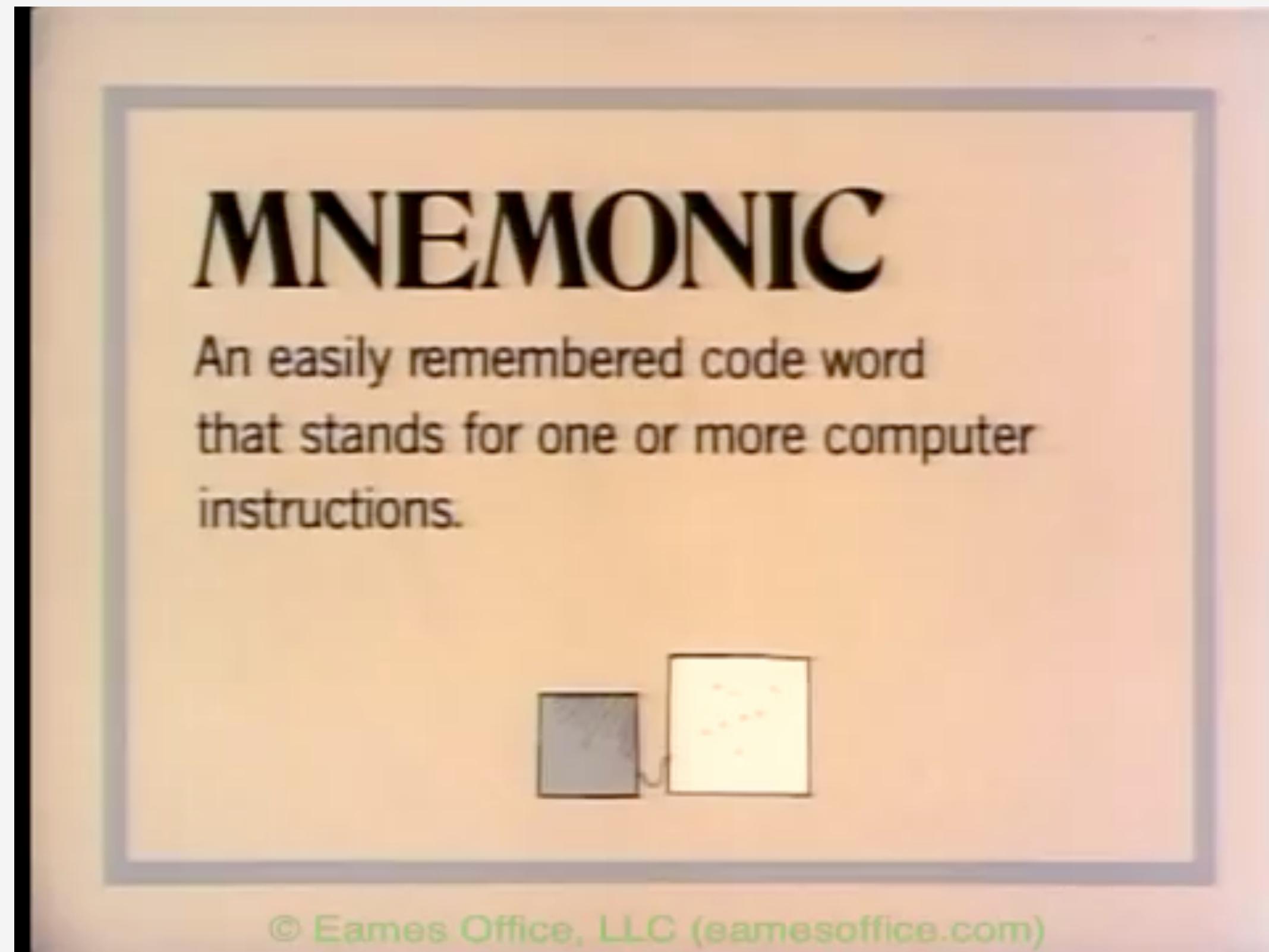


# CREATE A DIGITAL ARTWORK



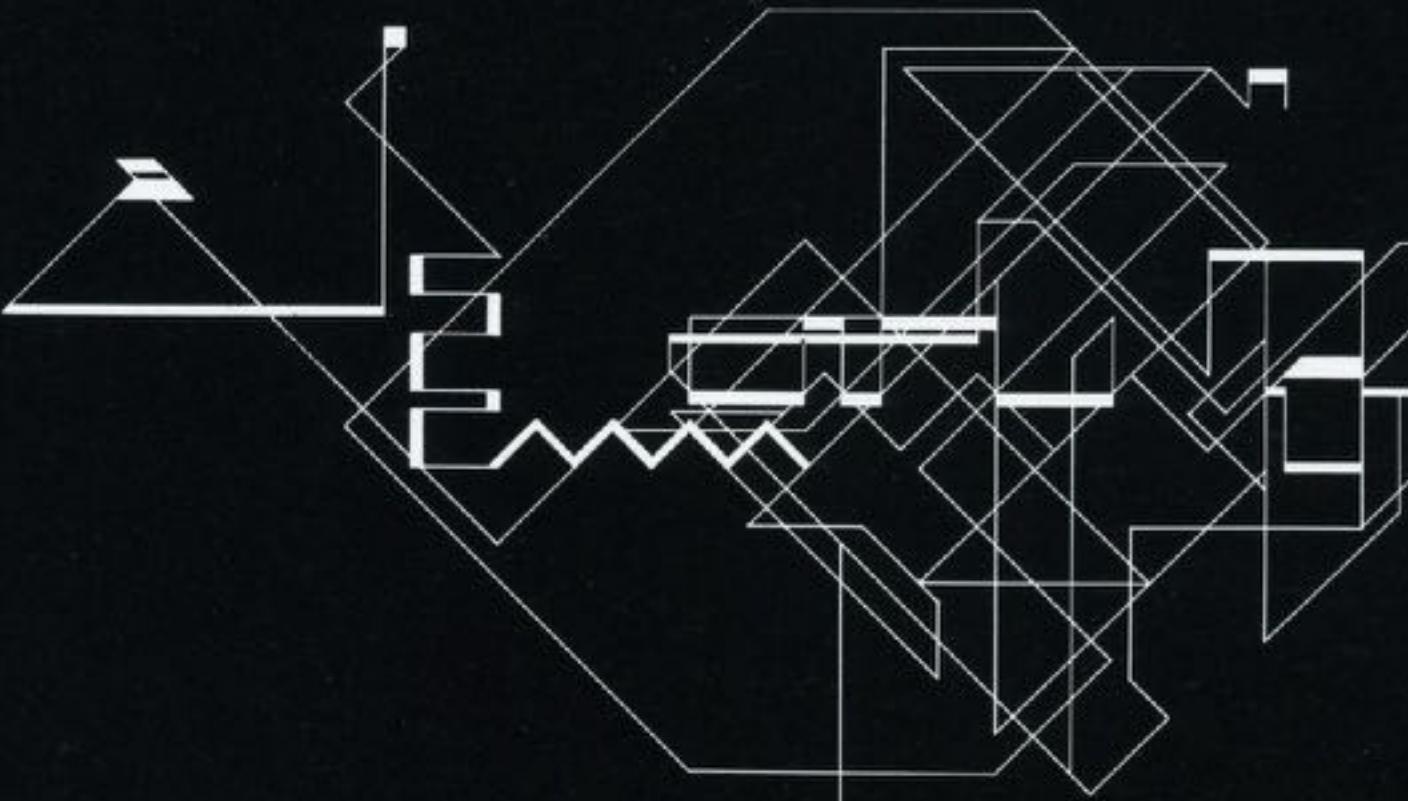
# A COMPUTER GLOSSARY

Charles and Ray Eames 1966

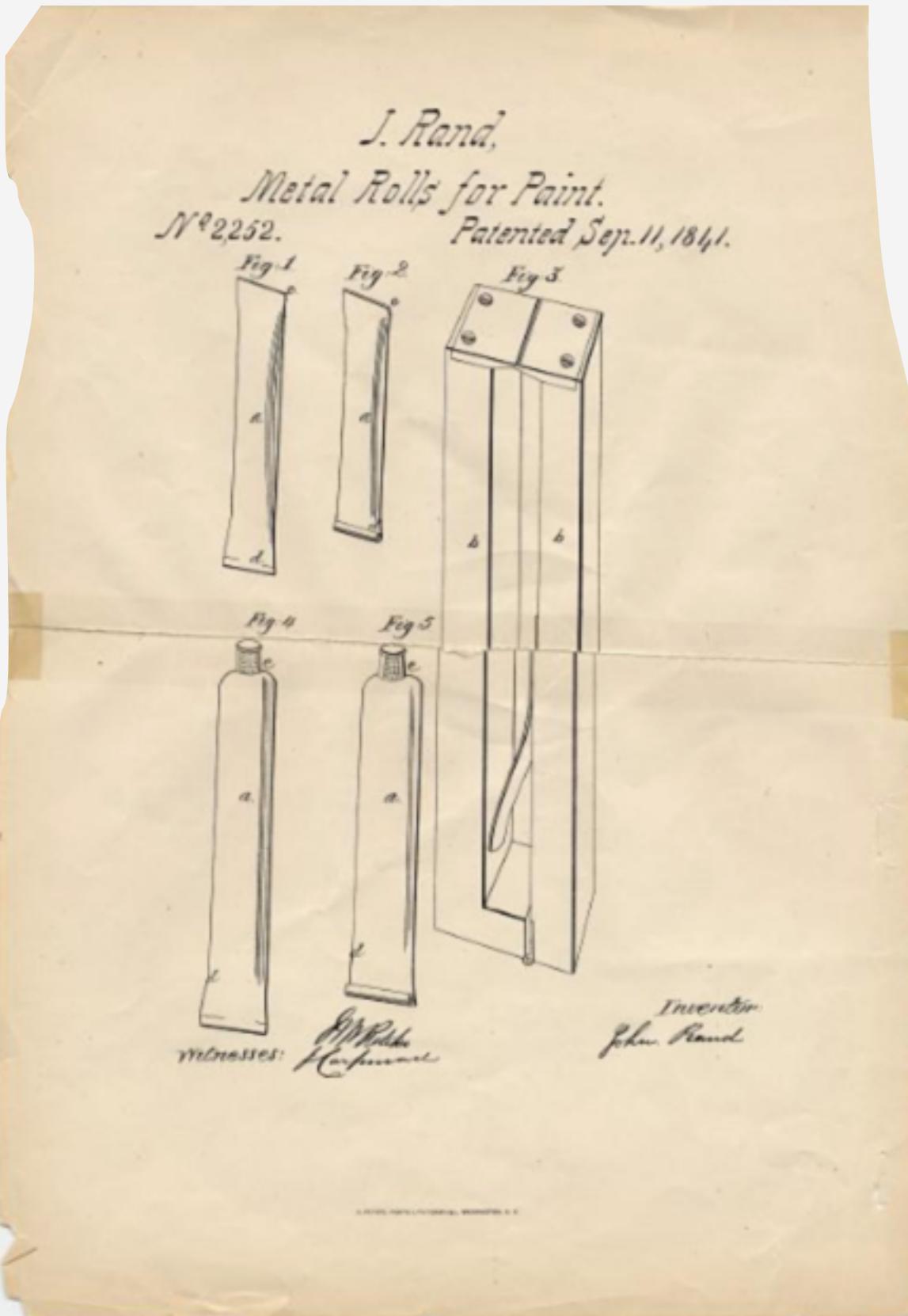


<https://www.youtube.com/watch?v=elgX6sPOqCY>

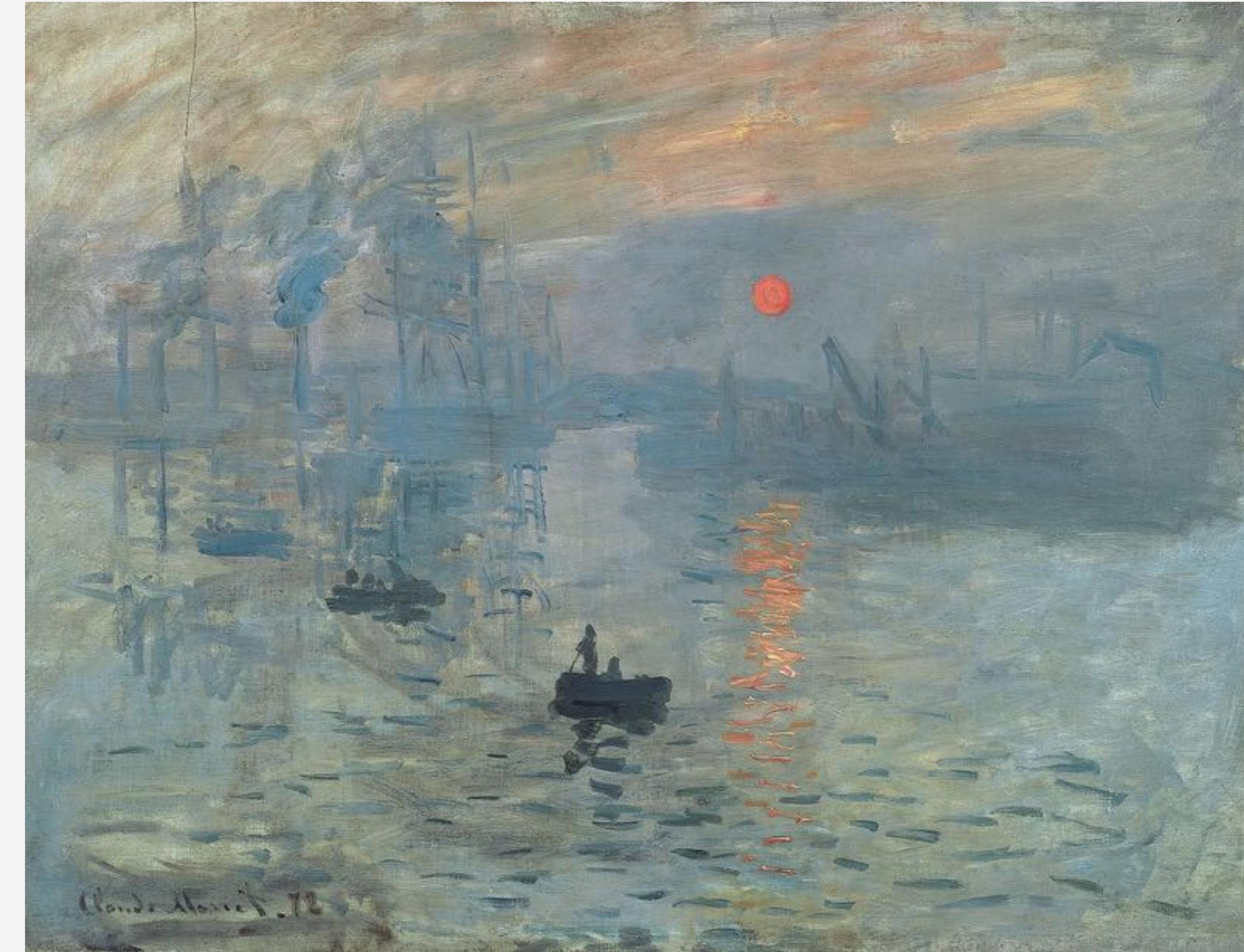
BUT WHAT DOES THAT HAVE TO DO WITH ART?



# ART AND TECHNOLOGY

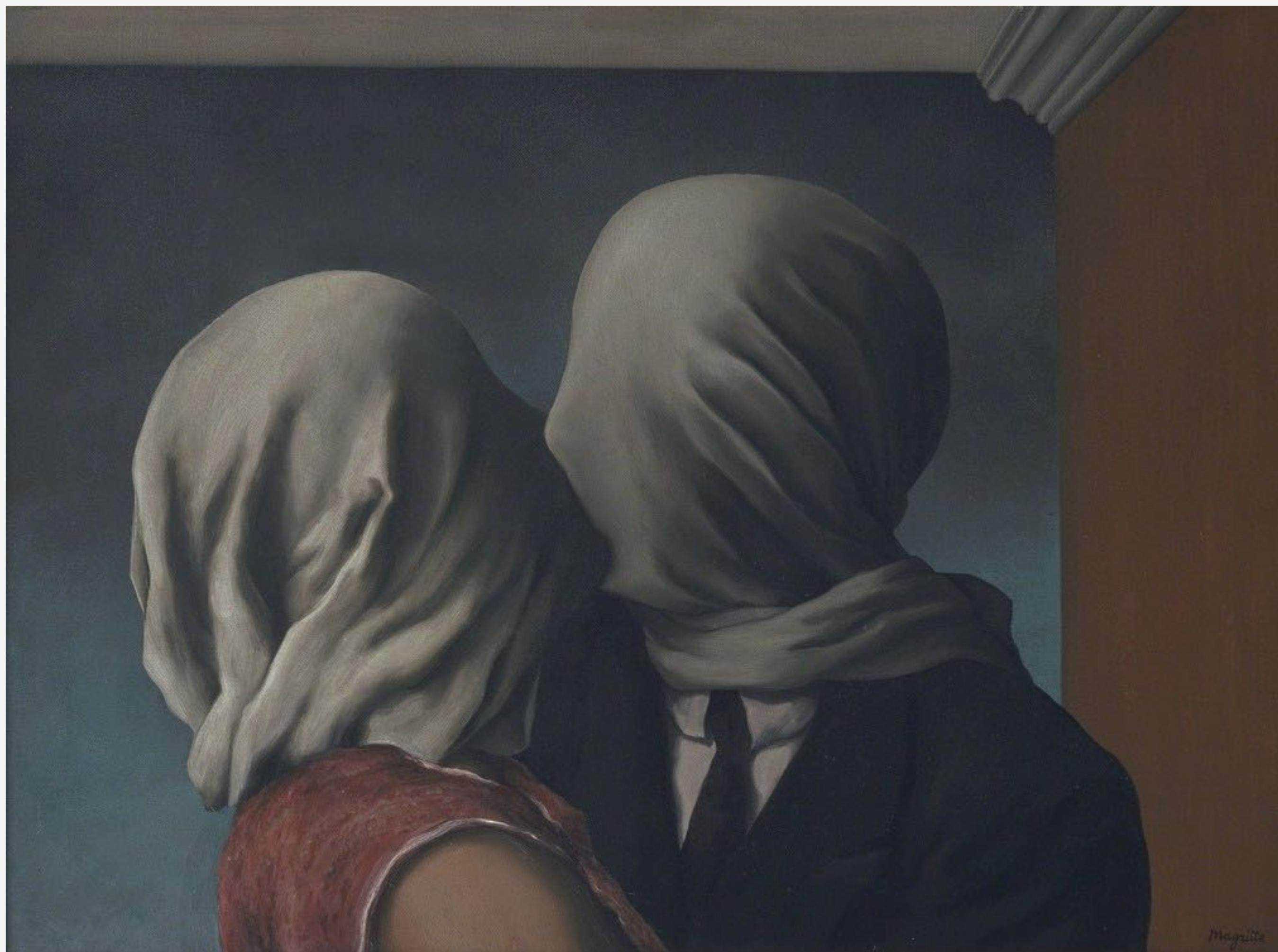


John Goffe Rand, Collapsible metal tube



Claude Monet, Impression, soleil levant

# VISUAL STORY TELLING



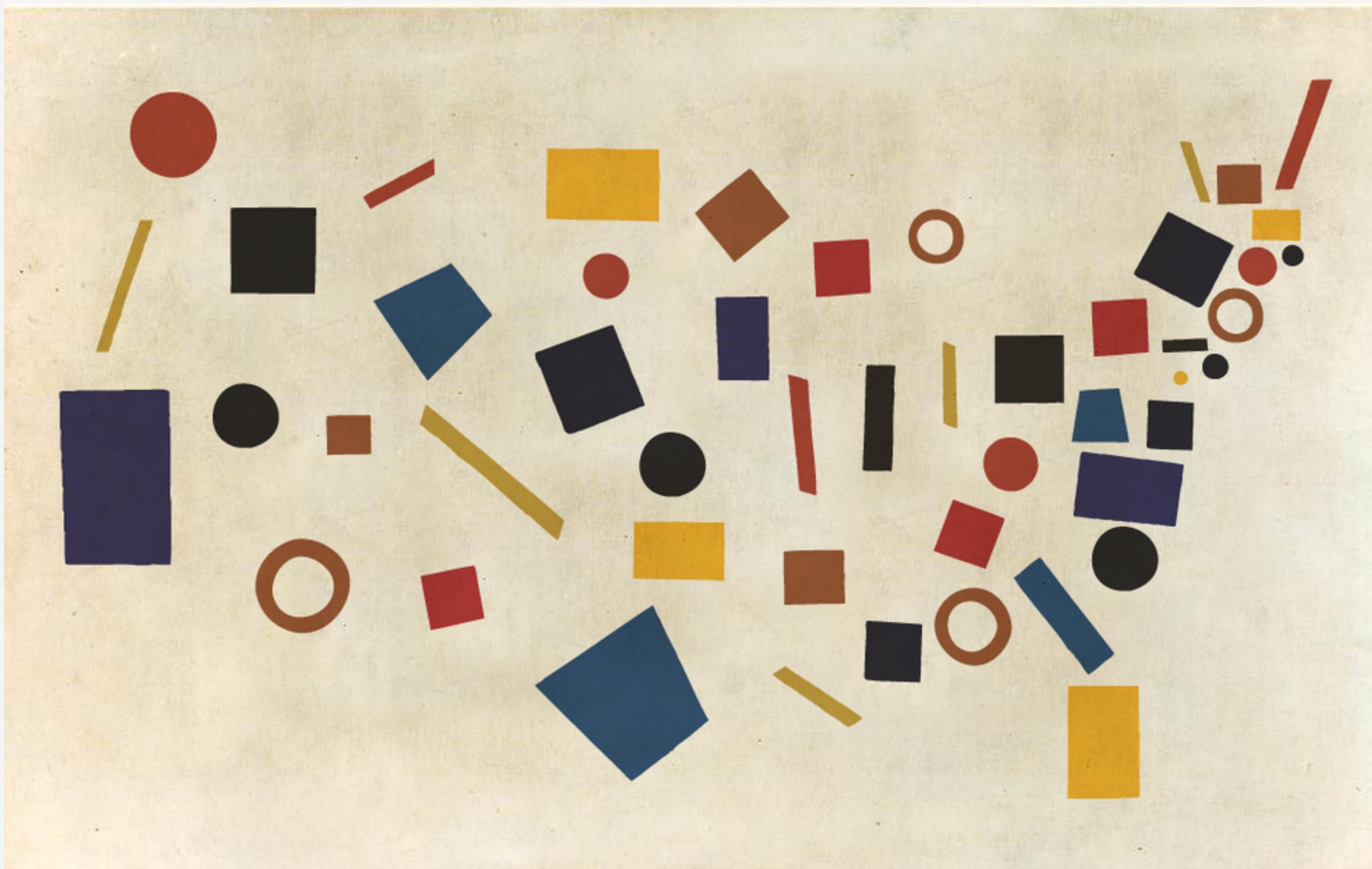
Rene Magritte



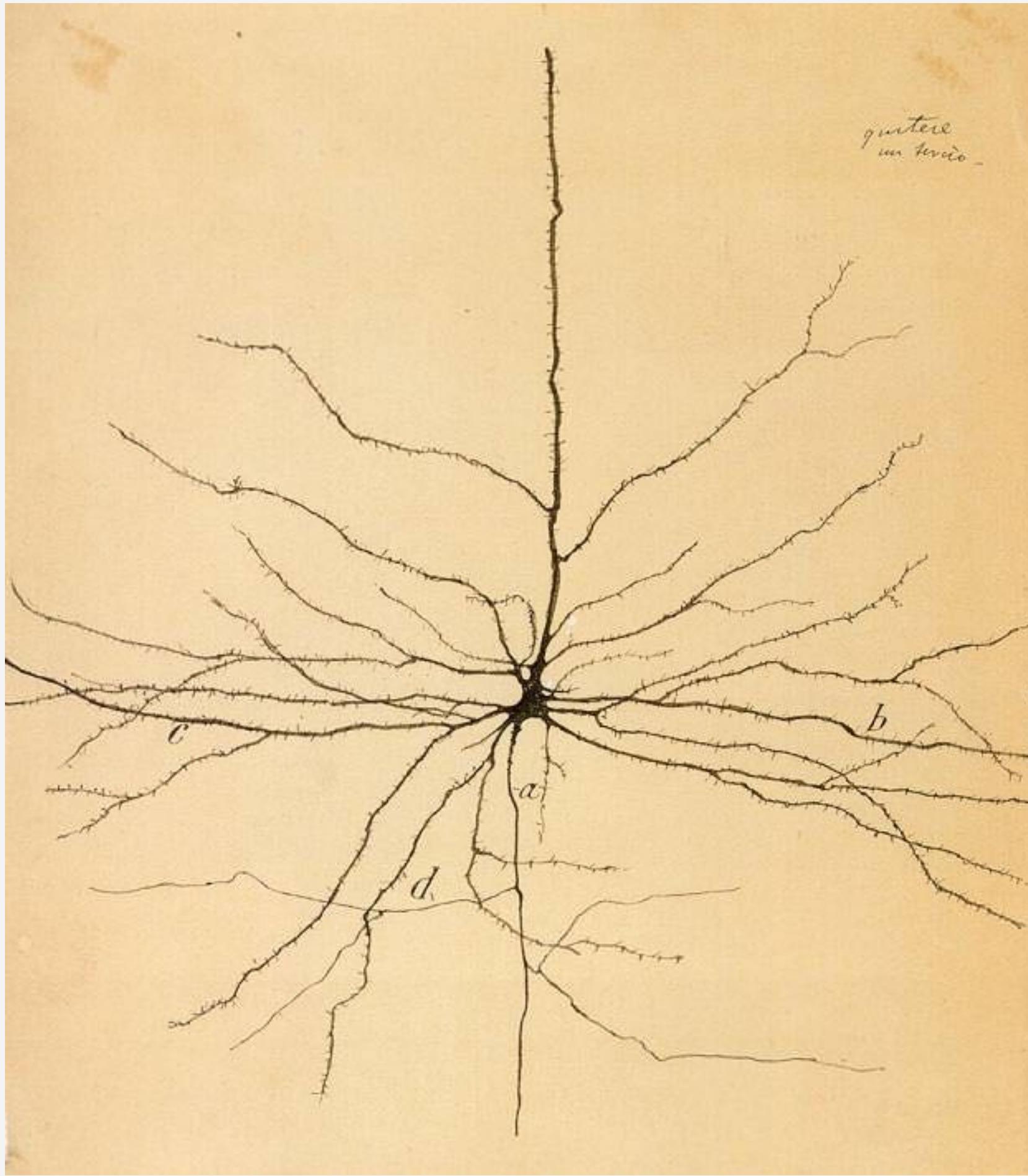
Anselm Kiefer



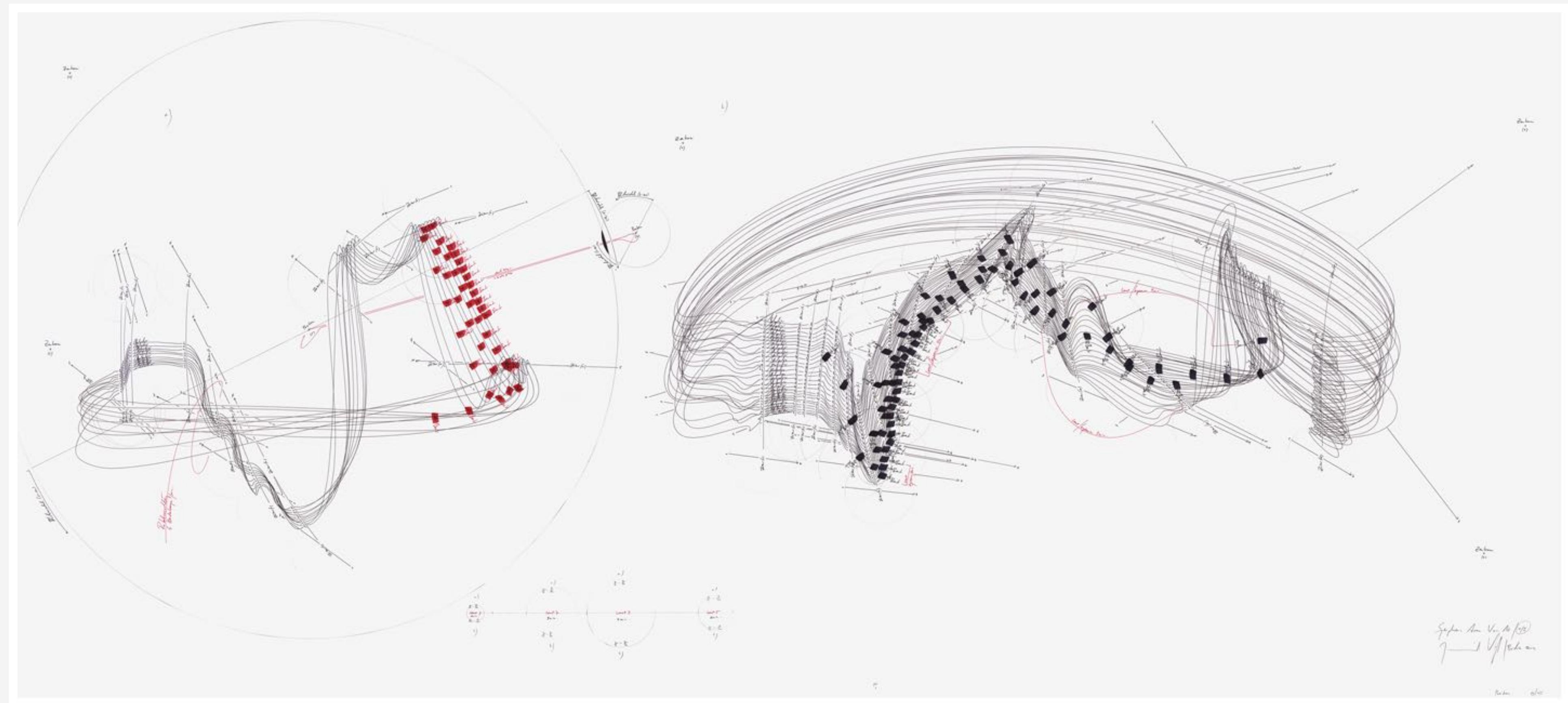
Luigi Serafini



Kazimir Malevich



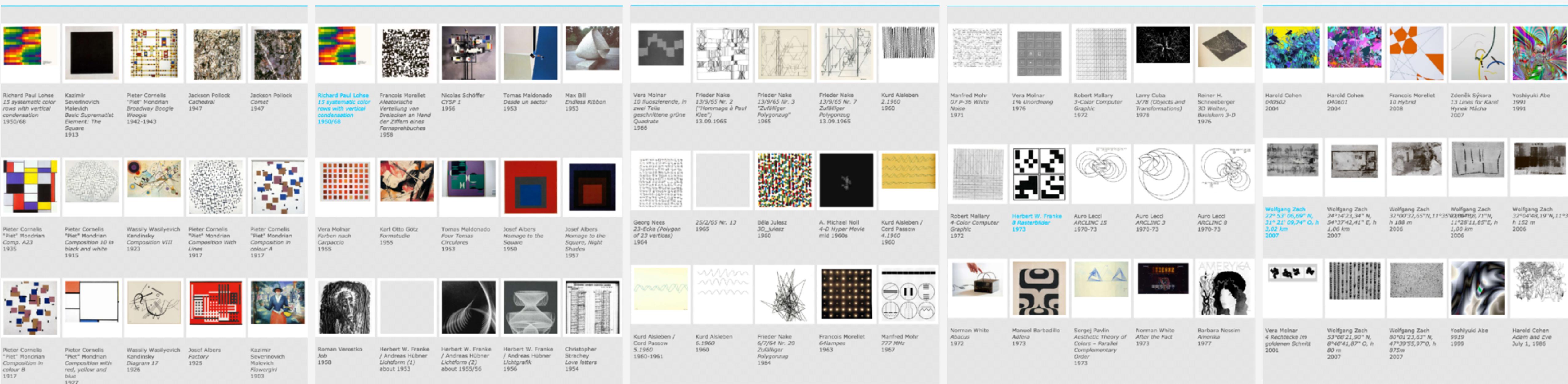
Santiago Ramón y Cajal



Jorinde Voigt

# BRIEF HISTORY OF COMPUTER ART

Computational and algorithmic art during the 20th century



Before 1950

1950 - 1960

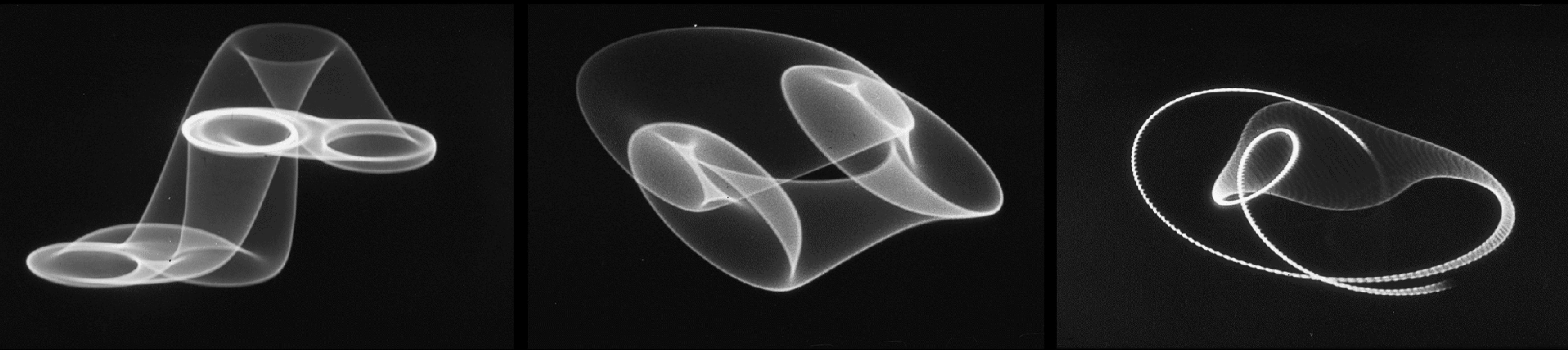
1960 - 1970

1970 - 1980

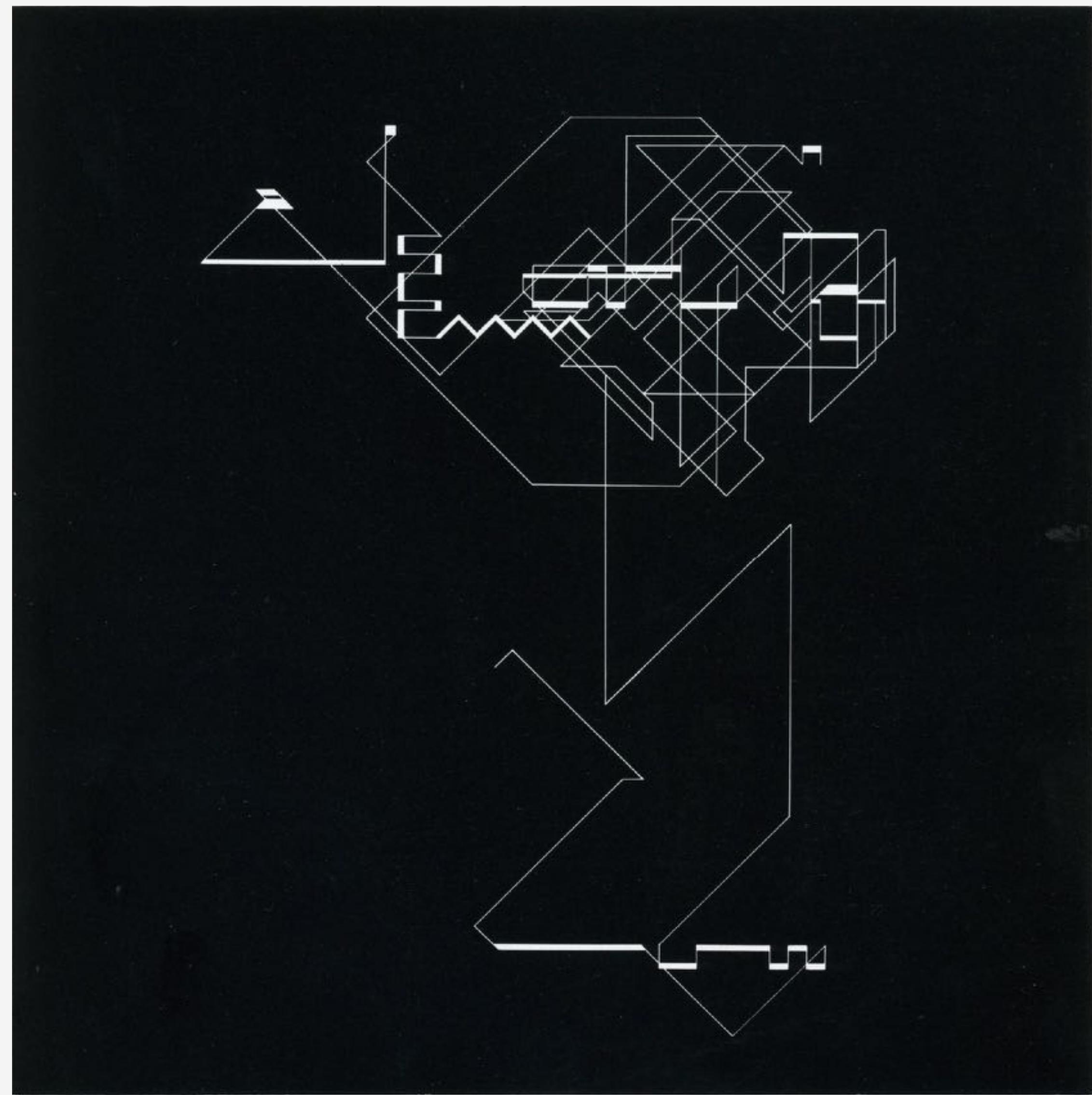
After 1980

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

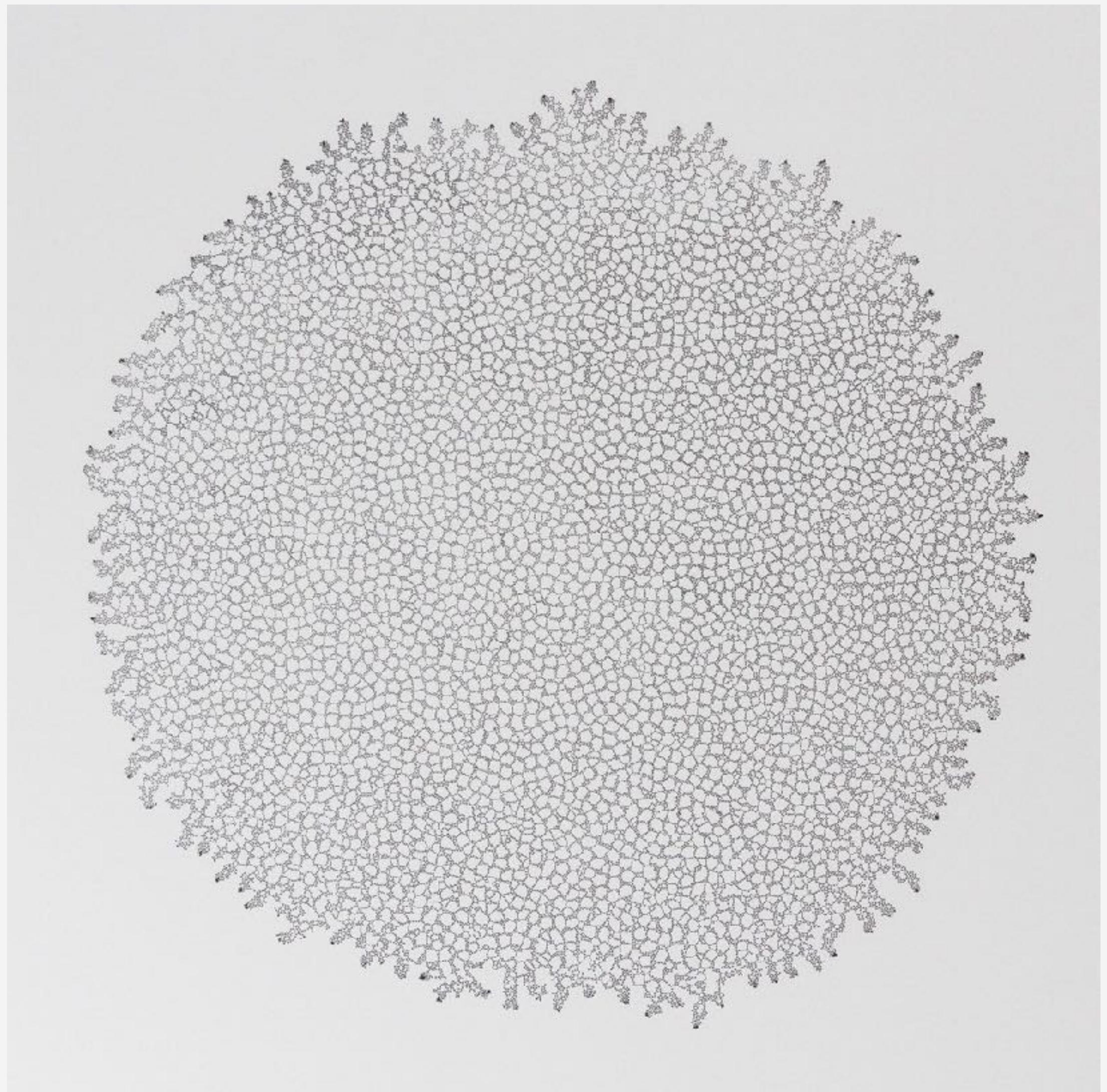
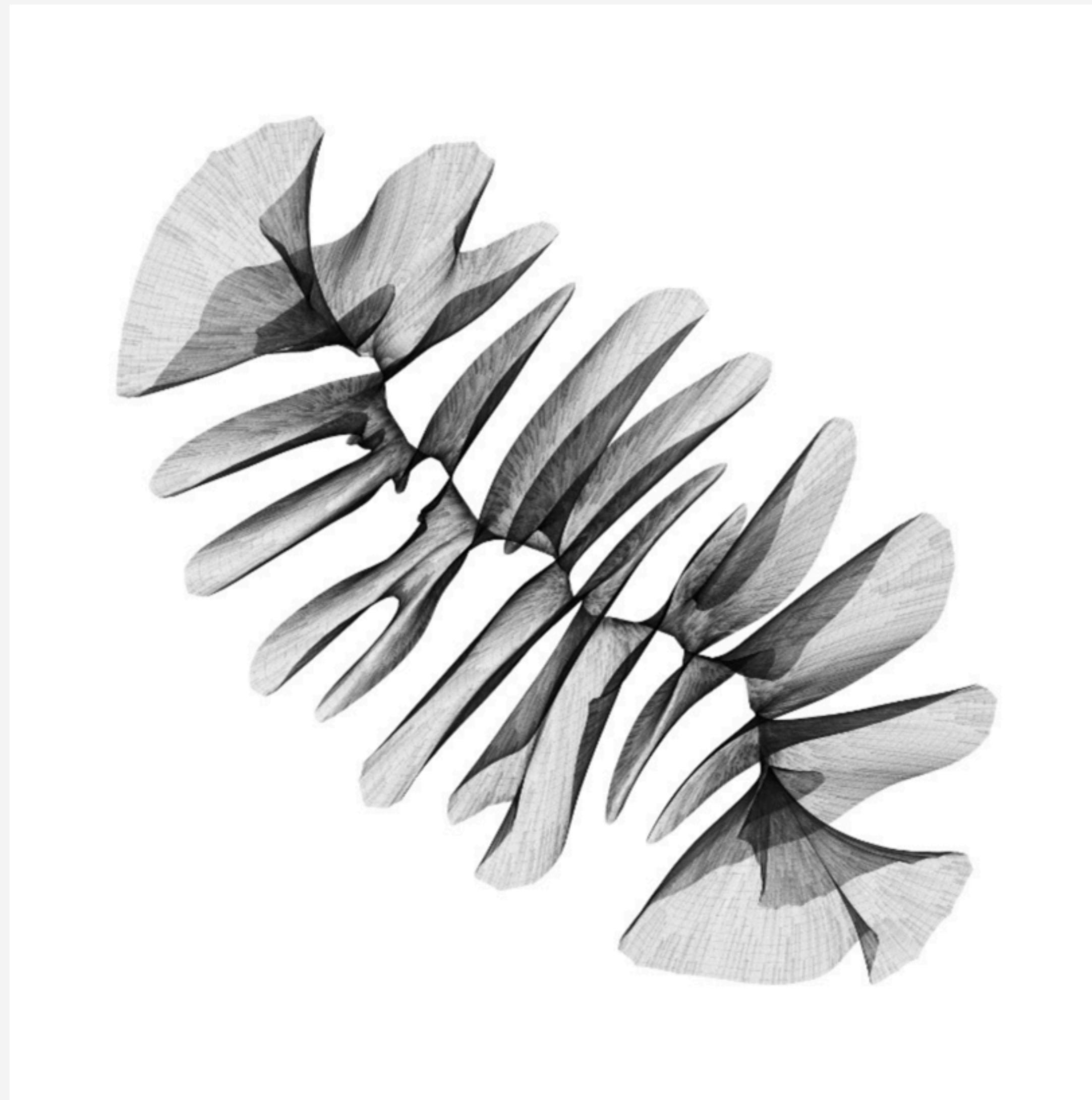
<http://digitalartmuseum.org/history>



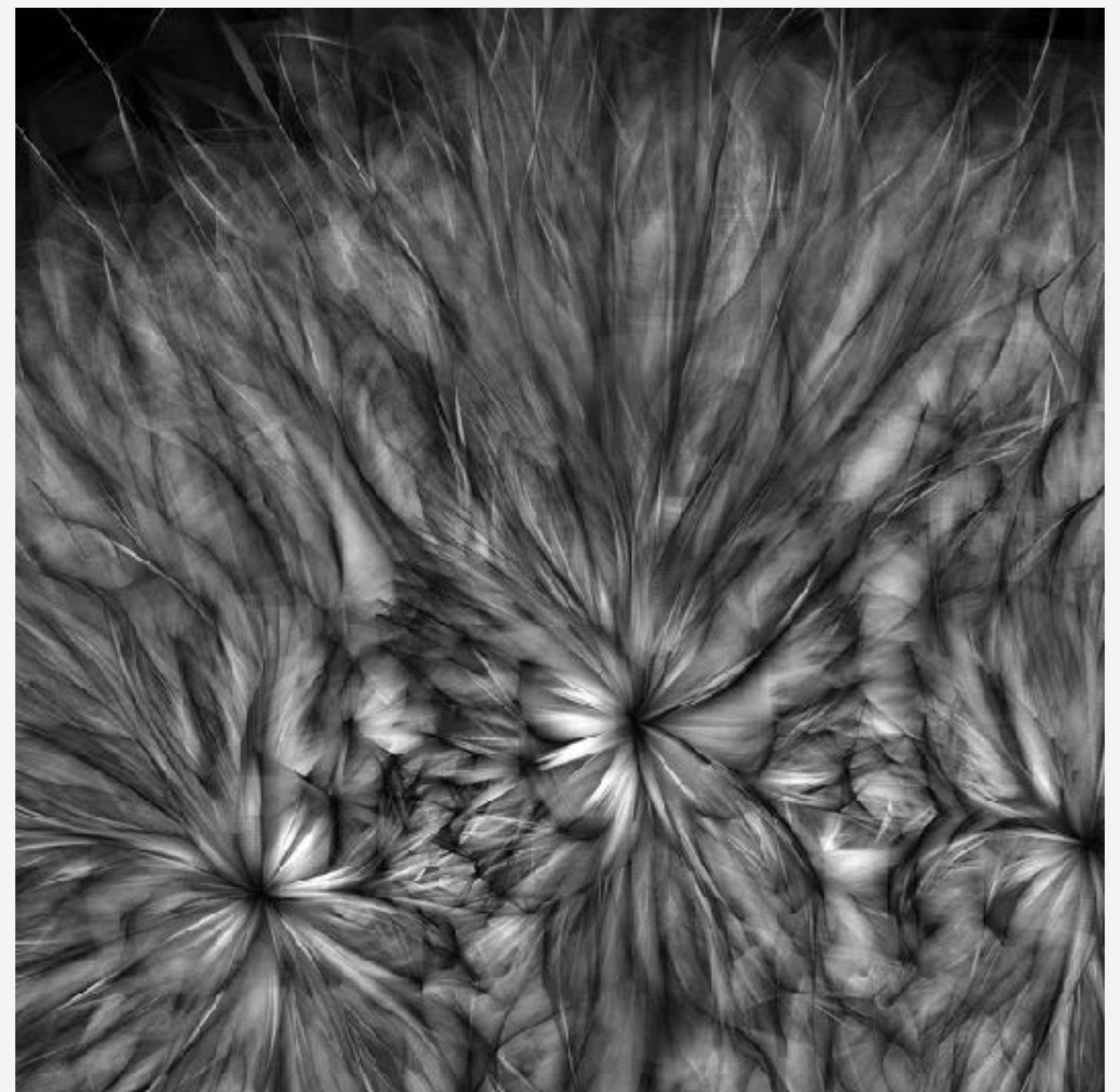
Herbert W. Franke



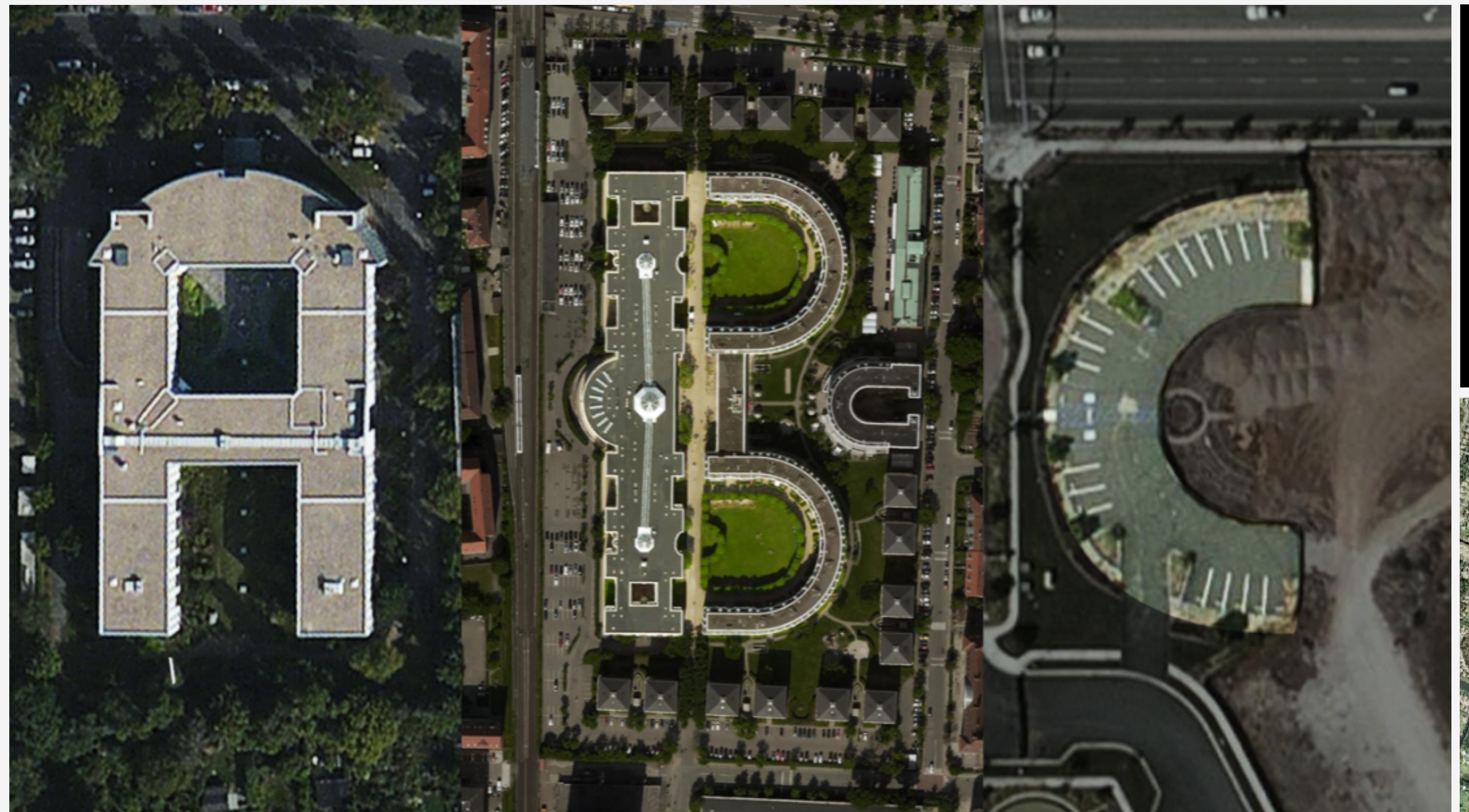
Manfred Mohr



Andreas Hoff



Casey Reas



KICKSTART THE  
PLATETRY SEARCH  
FOR LETTERFORMS



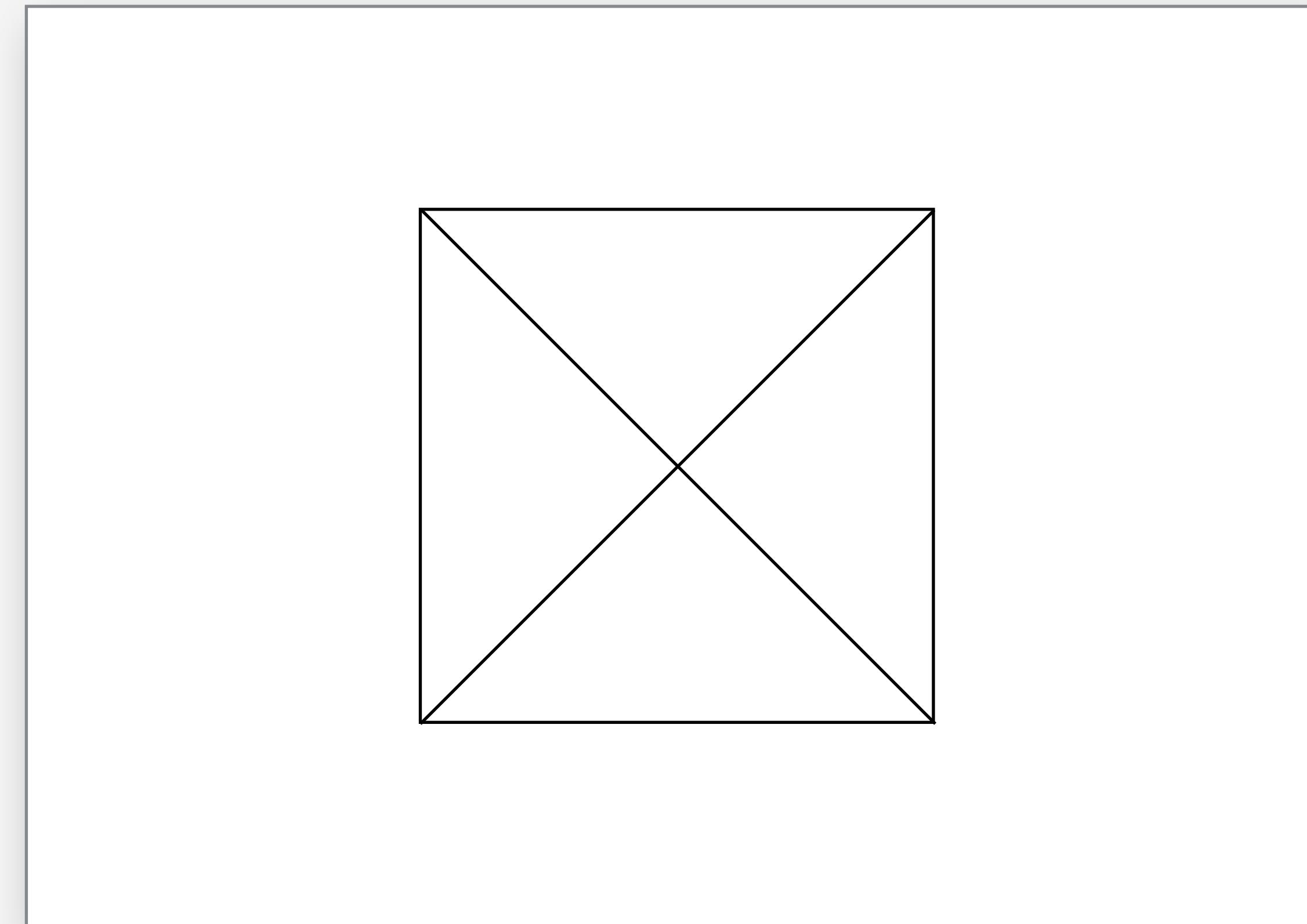
Benedikt Groß

# PROCEDURAL DRAWING LANGUAGE

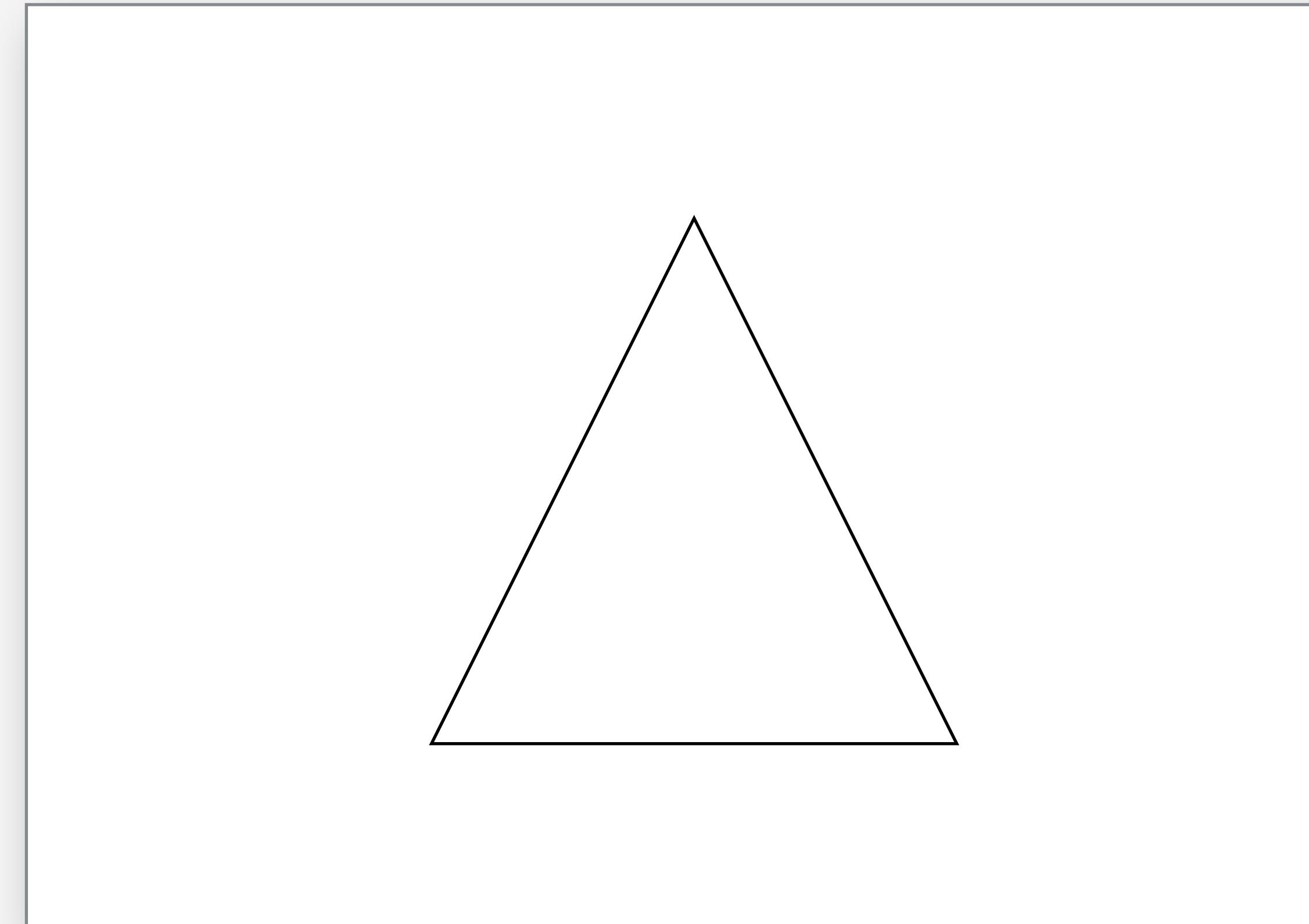
## **Class exercise**

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

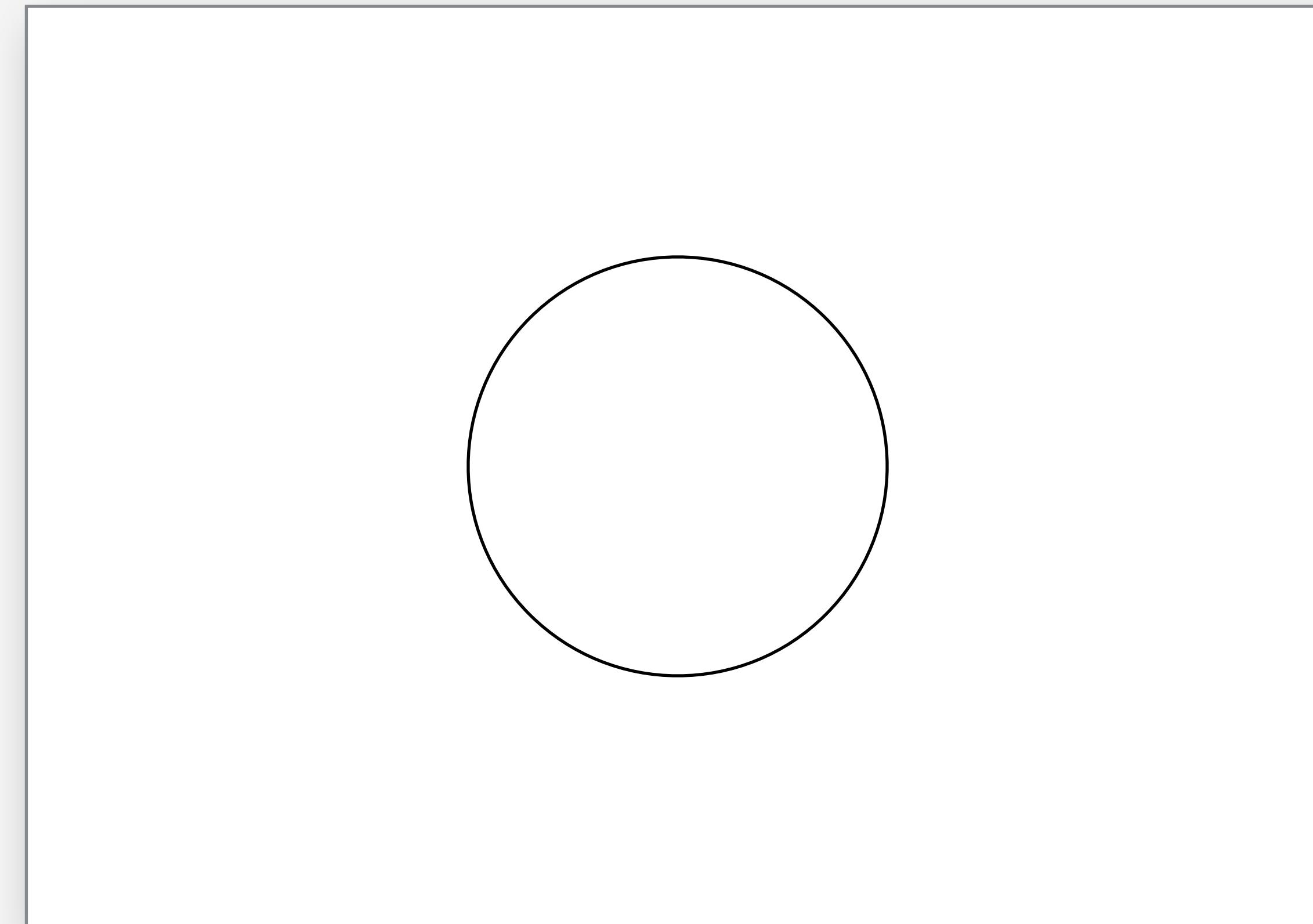
# DRAWING LANGUAGE



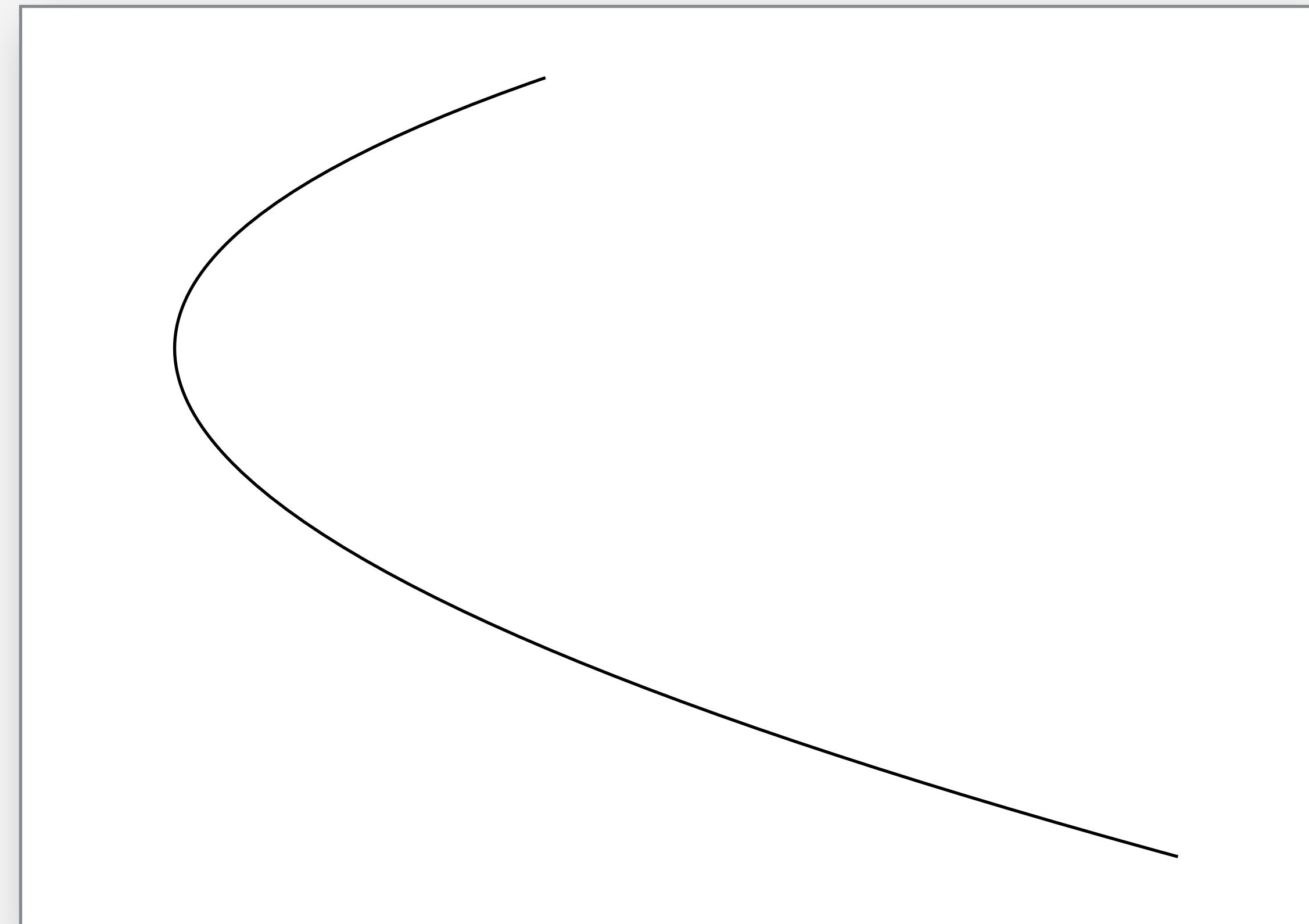
# DRAWING LANGUAGE



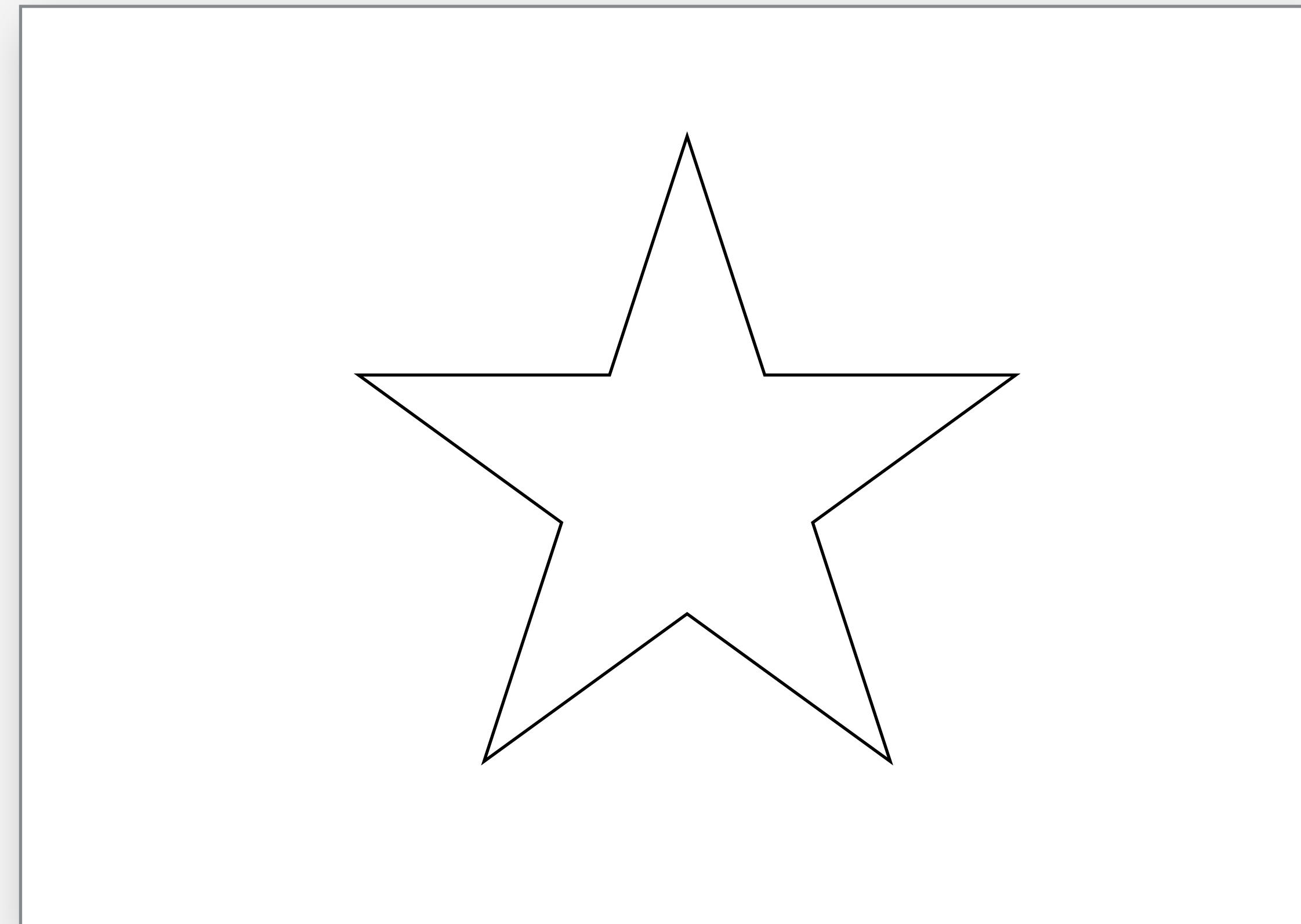
# DRAWING LANGUAGE



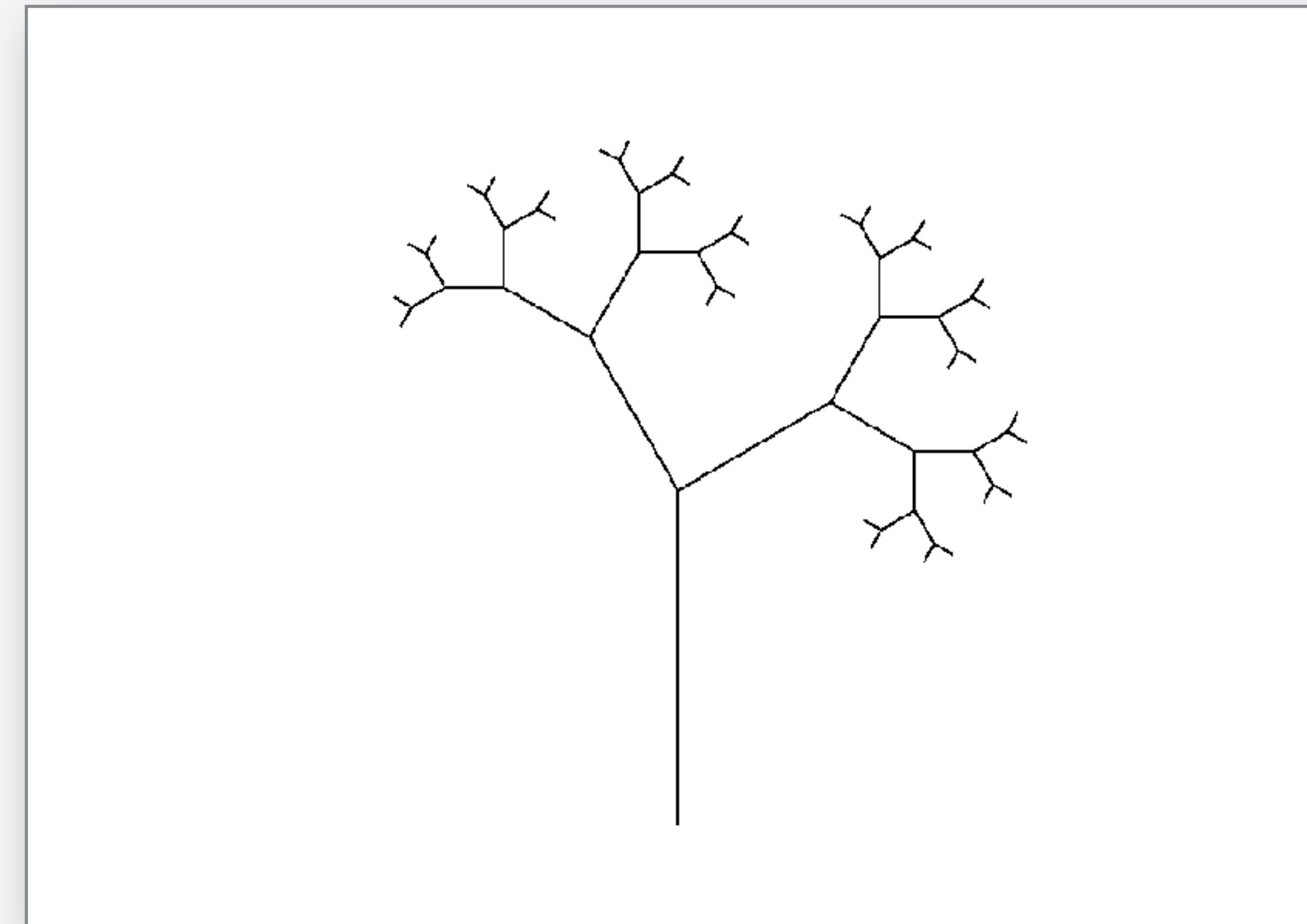
# DRAWING LANGUAGE



# DRAWING LANGUAGE



# DRAWING LANGUAGE



# THIS IS OUR NEW WORKSPACE:

The screenshot shows the p5.js Reference website. It features a large pink p5.js logo at the top left. Below it is a navigation bar with links for Processing, p5.js, Processing.py, Processing for Android, and Processing for Pi. The main content area is titled "Reference" and includes a search bar labeled "Search the API". A sidebar on the left contains links for Home, Download, Start, Reference, Libraries, Learn, Examples, Books, and Community. The Reference section lists categories like Color, Environment, Lights, Camera, Structure, Constants, Events, Math, Transform, DOM, IO, Rendering, Typography, Data, and Image. At the bottom, there are links for Forum, GitHub, and Twitter.

The screenshot shows the p5.js Web Editor. It has a header with the p5.js logo and a search bar. The main area is divided into two sections: "sketch.js" on the left containing a code editor with the following code, and "Preview" on the right showing a canvas output. The code in the editor is:

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

function draw() {
  background(220);
}
```

The screenshot shows the CodePen website. The top navigation bar includes links for Log in or Sign up, CREATE, YOUR, EXPLORE, GROW, and more. The main content area is titled "Picked Pens" and displays a grid of cards for various code snippets. One card is highlighted with a larger preview. On the right side, there are sliders for "rotation axis" and "rotation angle".

[p5js.org/reference](https://p5js.org/reference)

[editor.p5js.org](https://editor.p5js.org)

[codepen.io](https://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 EXERCISES

Do one of the following:

- Create a processing sketch which uses all 2D drawing commands.
- Create a composition which expresses the word "CONTACT" using only the arc() drawing command
- Pick an object in the room and create an abstract visualization of it.

# HOMEWORK

## A. Program a self portrait

Please Program a self portrait in p5.

Be mindful of details and make purposeful visual decisions.

## B. Watch:

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