

assignment1

INTERACTIVE DRAWING TOOL

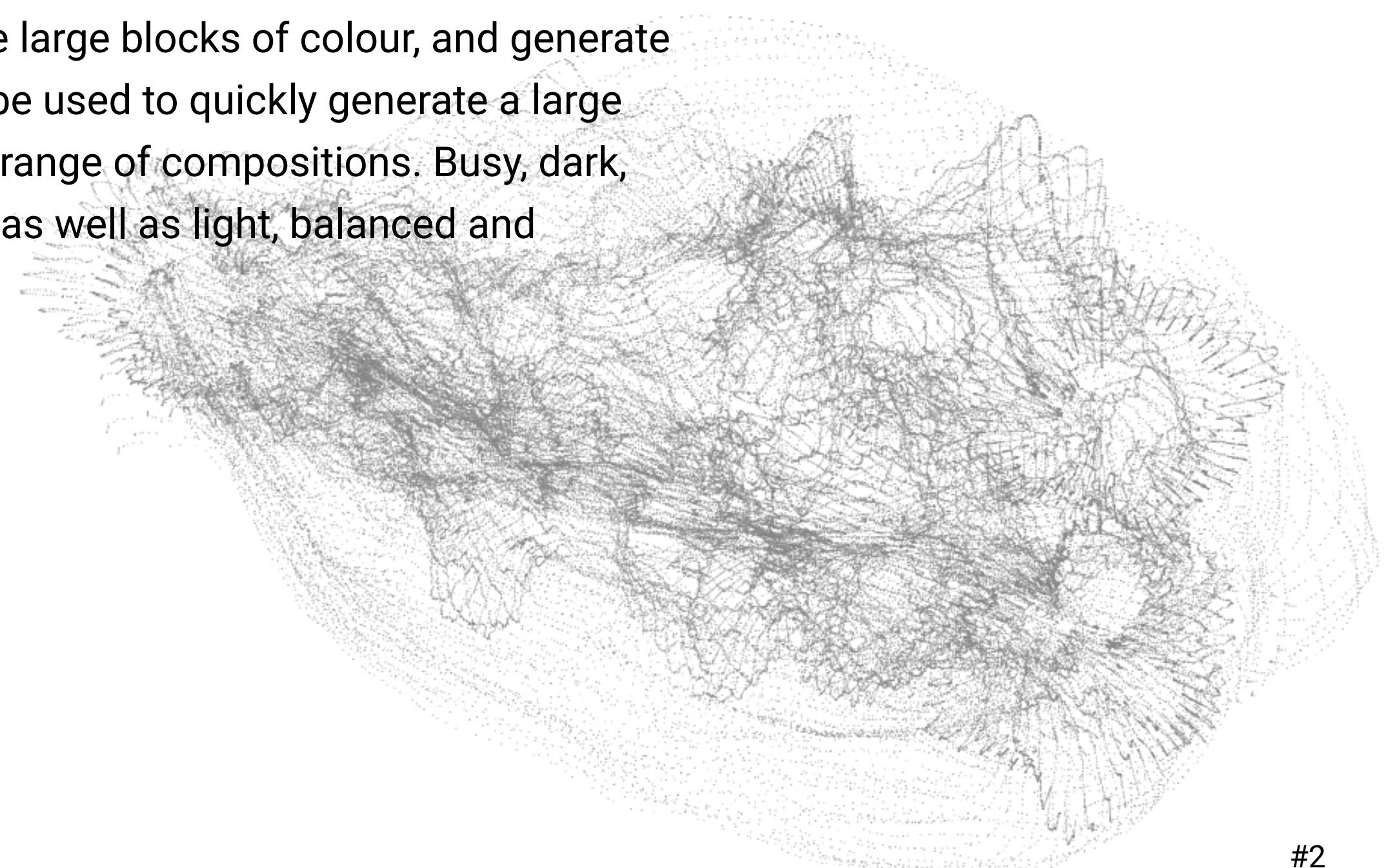
Computational Prototyping
RMIT university

william wallis s3773723
April 2022

//GRAPHIX GEN 3000

written description

Graphix gen 3000 (gg3000) is an interactive drawing tool that generates geometric line art. The program has five brushes that can be interacted with to create linear geometric forms, pointilated forms, broken line geometric forms, can be used to create large blocks of colour, and generate forms from past images. Gg3000 can be used to quickly generate a large number of graphic images with a wide range of compositions. Busy, dark, and colourful graphics can be created, as well as light, balanced and elegant graphics.



//GRAPHIX GEN 3000

program operation

To start the program hit **ENTER**.

There are 5 brushes that can be interacted with and selected using keys 1-5

Key == 1: round line generating tool

Key == 2: dot generating tool

Key == 3: broken line generating tool

Key == 4: opaque rectangular colour squares*

Key == 5: generative image brush

***SHIFT:** shift sets the x1 and y1 points of the rectangle,

dragging or clicking the mouse forms the rectangle

//All brushes are used through clicking and dragging

with **LEFT CLICK**]

Additional interactions include:

UP_ARROW: Increases noise in brushes 1 – 3

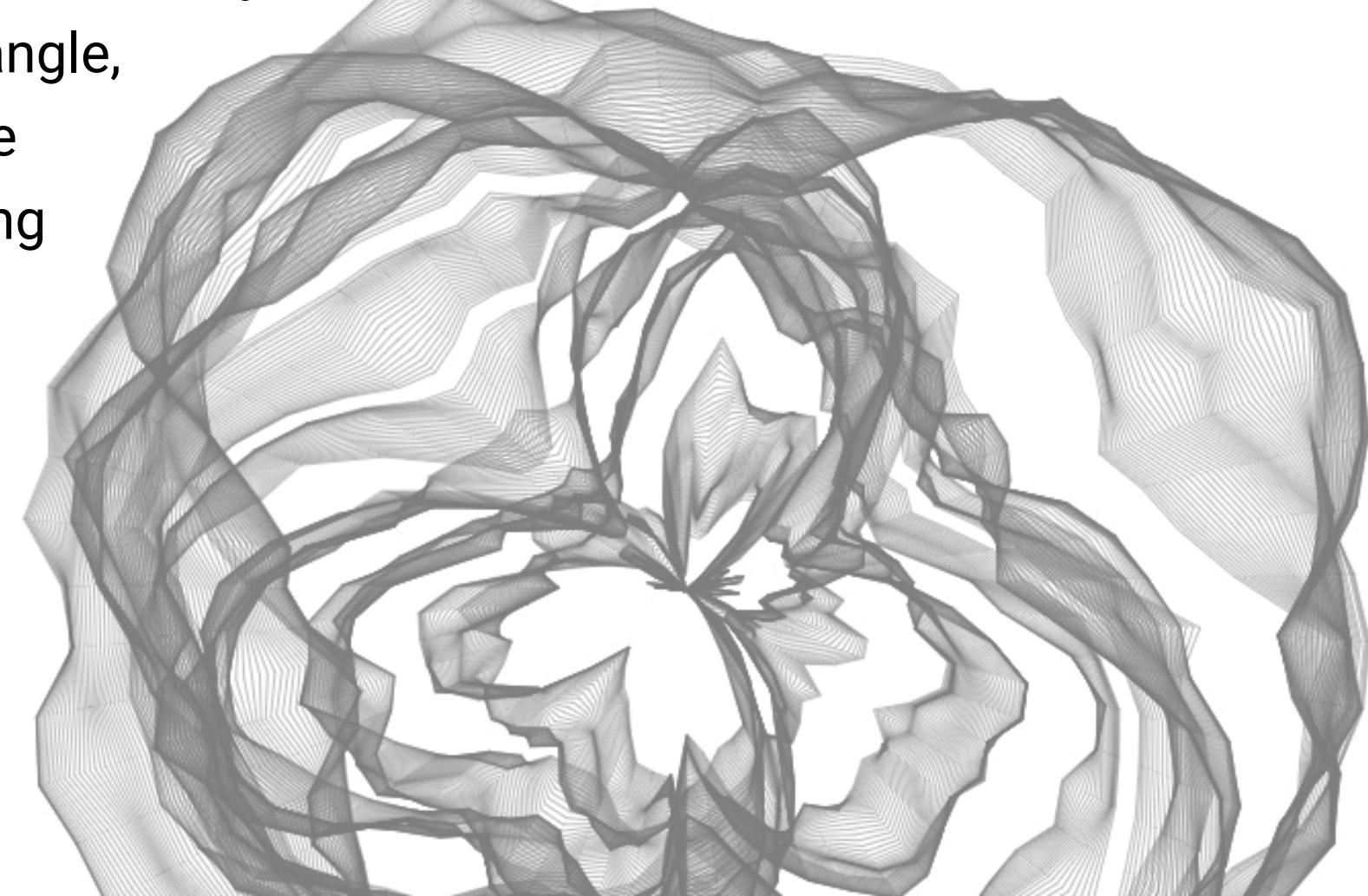
DOWN_ARROW: Decreases noise in brushes 1 – 3

Key == w: increases brush size of brushes 1 – 3

Key == s: decreases brush size of brushes 1 – 3

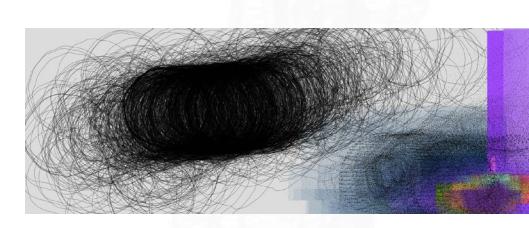
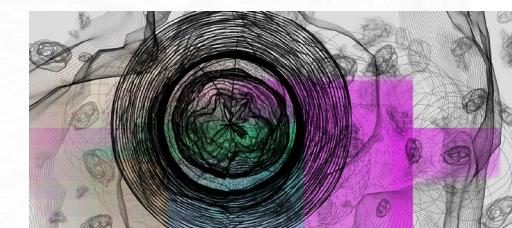
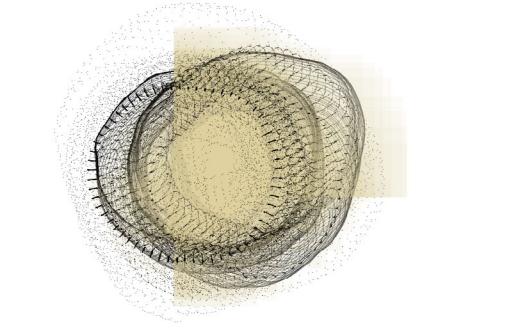
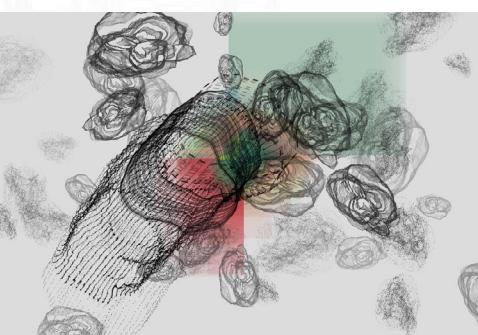
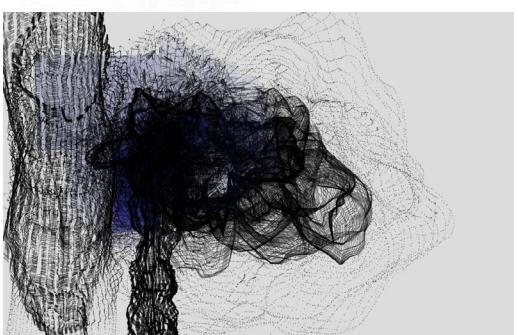
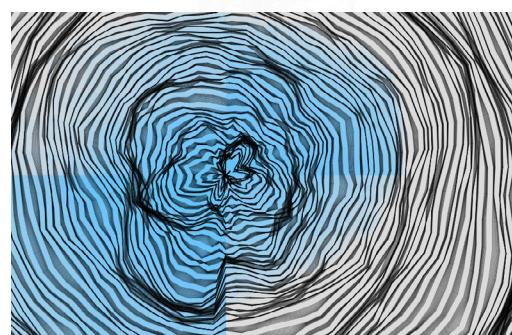
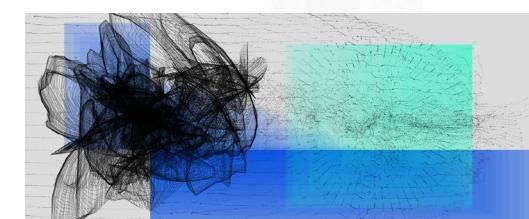
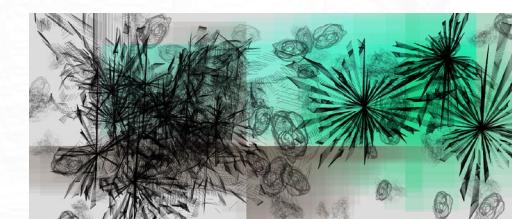
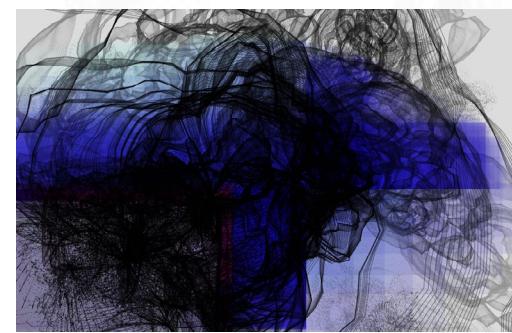
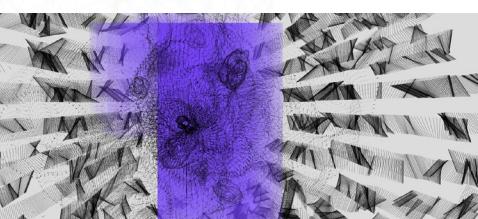
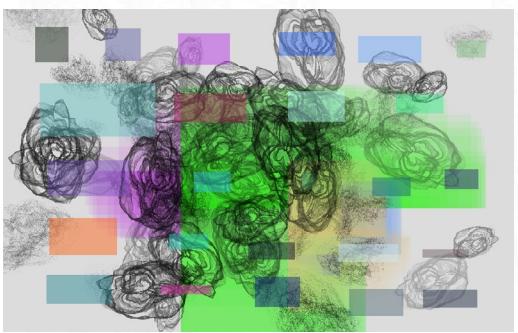
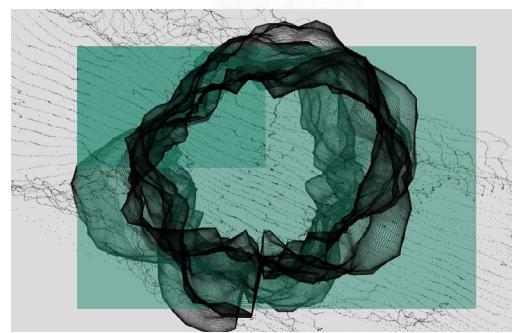
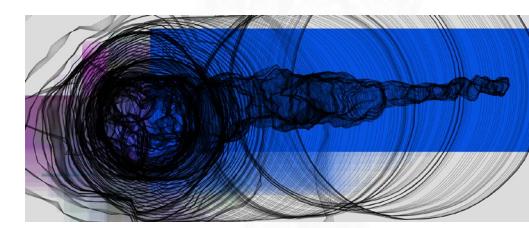
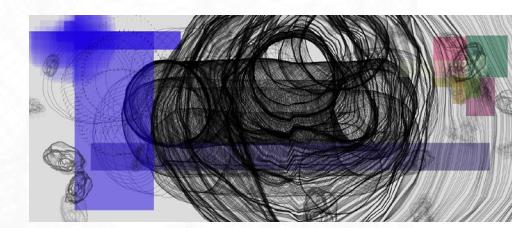
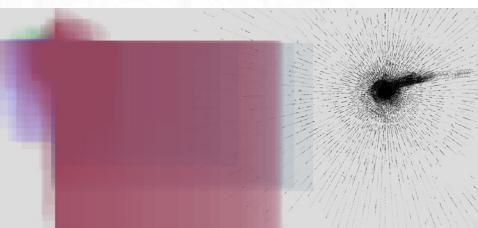
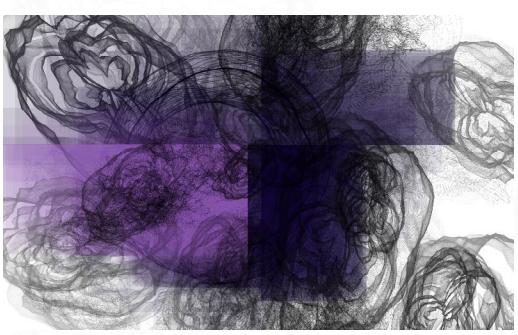
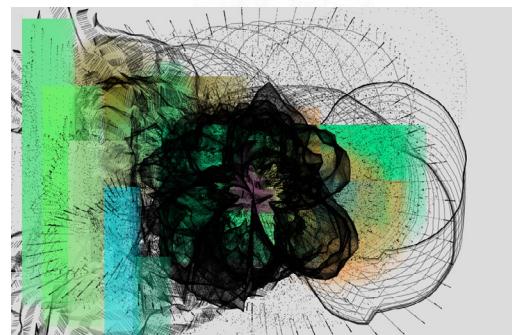
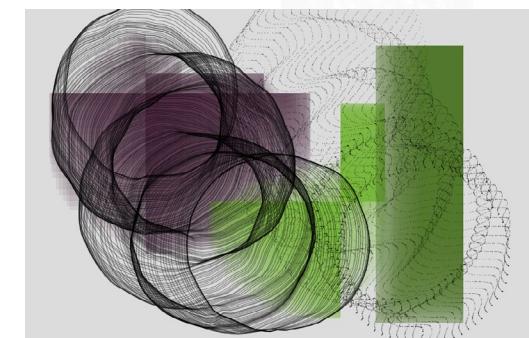
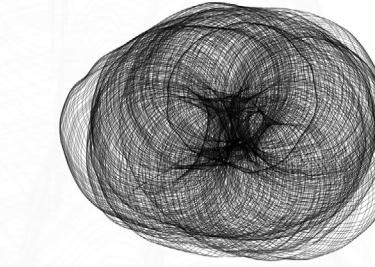
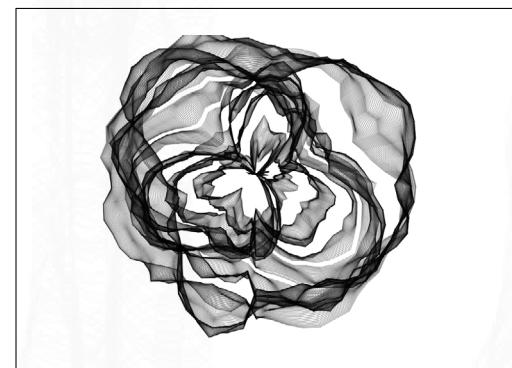
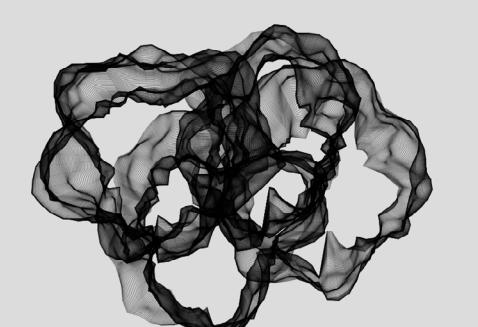
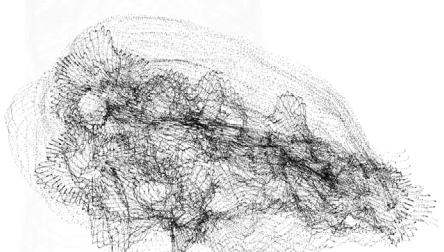
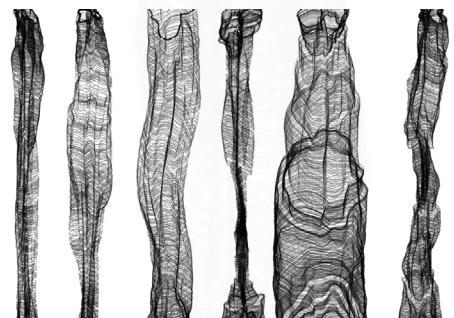
DELETE or BACKSPACE: clears the canvas

Key == z: saves the canvas



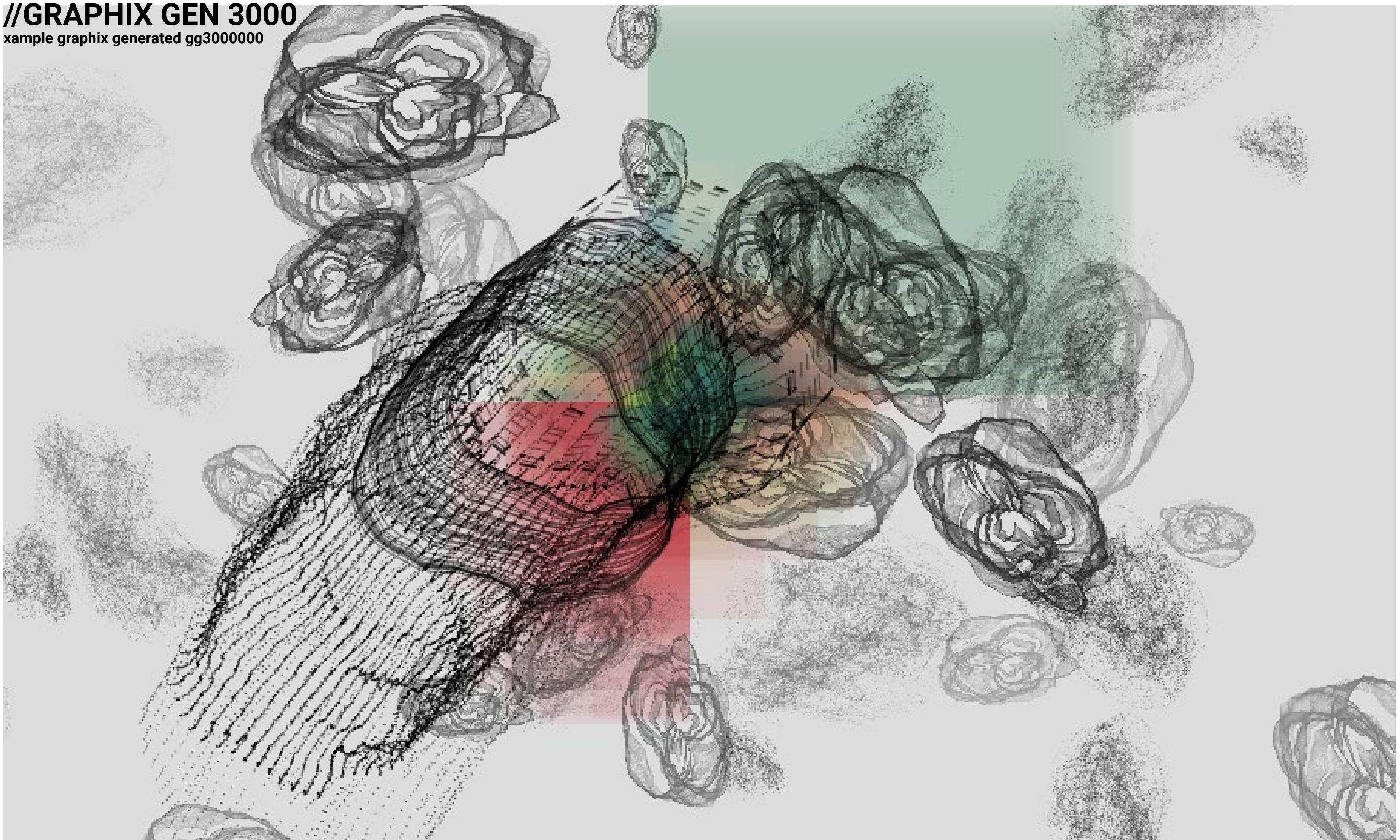
//GRAPHIX GEN 3000

xample graphix generated gg3000000000



//GRAPHIX GEN 3000

example graphix generated gg3000000

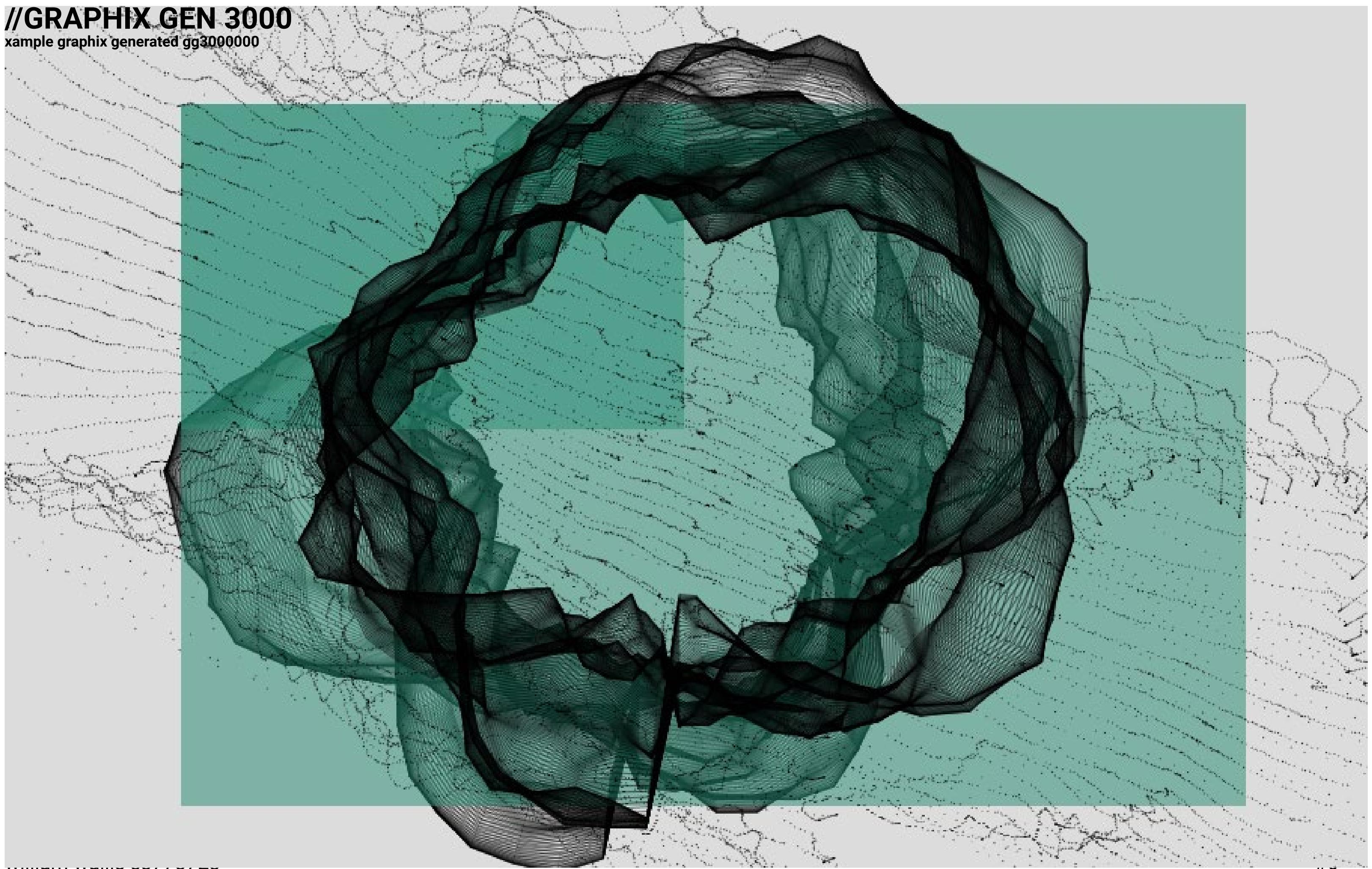


william wallis s3773723

#5

//GRAPHIX GEN 3000

xample graphix generated gg3000000



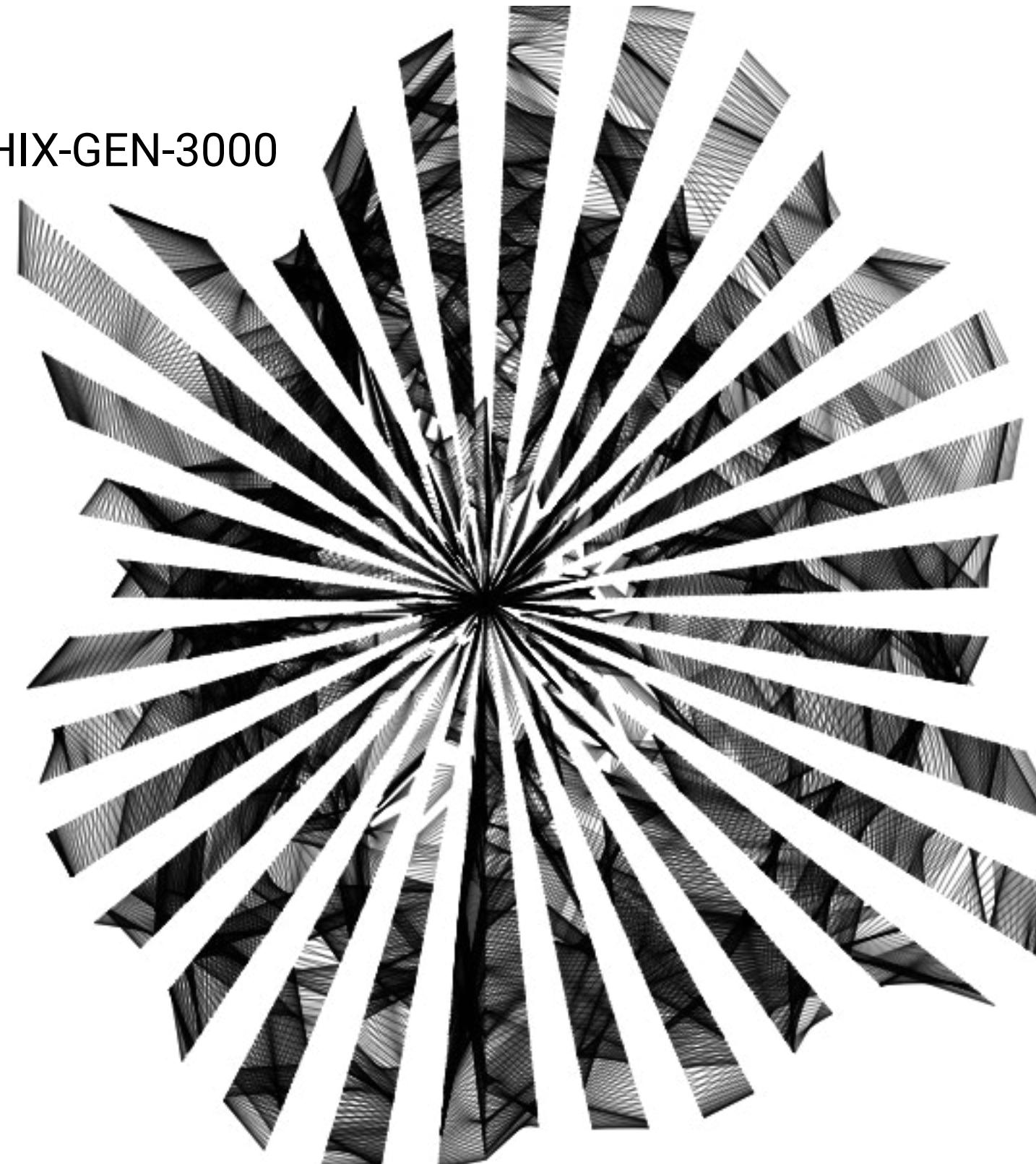
//GRAPHIX GEN 3000

original P5.js code with descriptive comments

Code + files can be found @: <https://github.com/wi11wa11/GRAPHIX-GEN-3000>

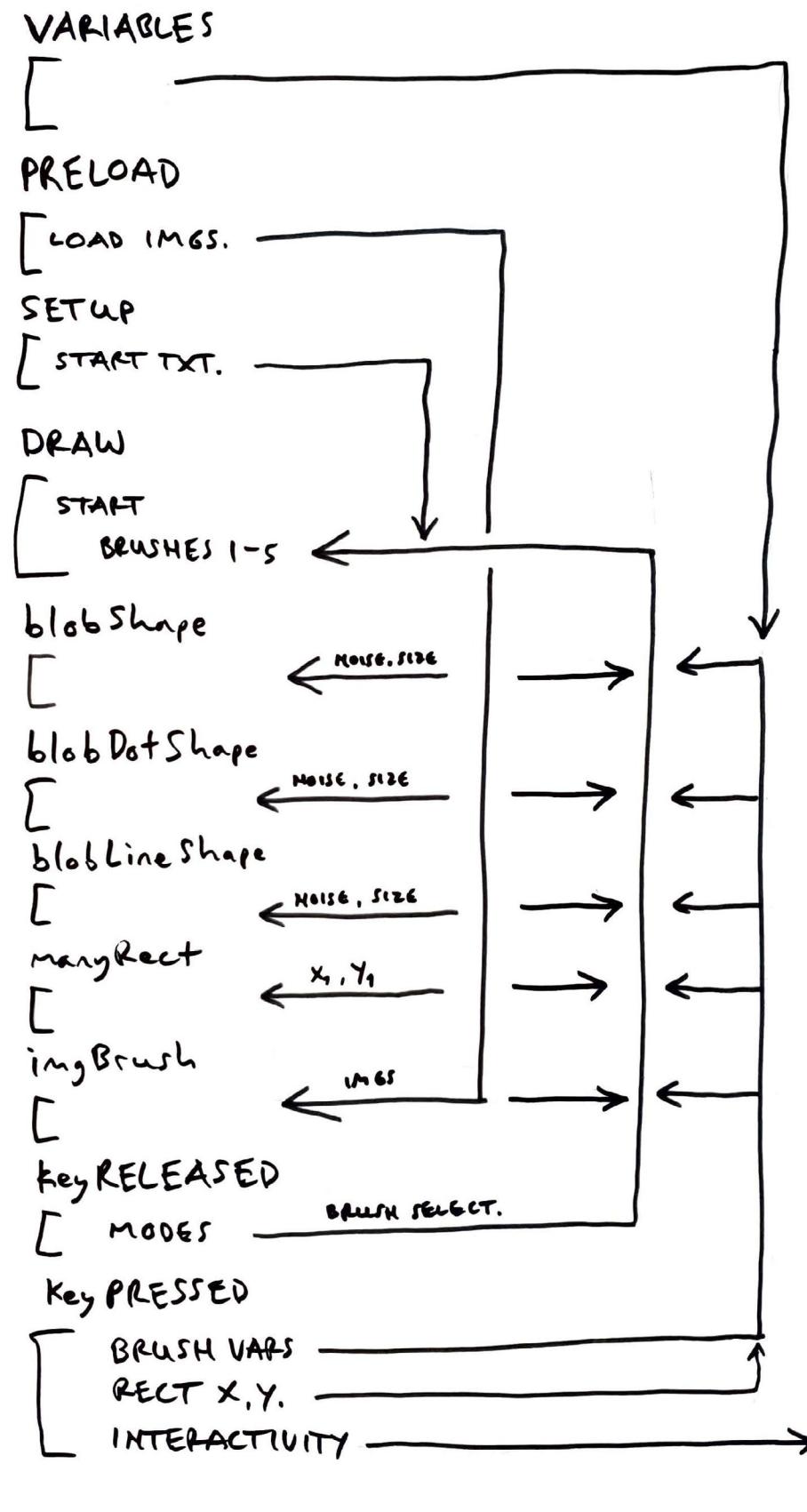
P5 sketch @: <https://editor.p5js.org/willwall/sketches/l9qdqF71E>

Github Page @: <https://wi11wa11.github.io/GRAPHIX-GEN-3000/>

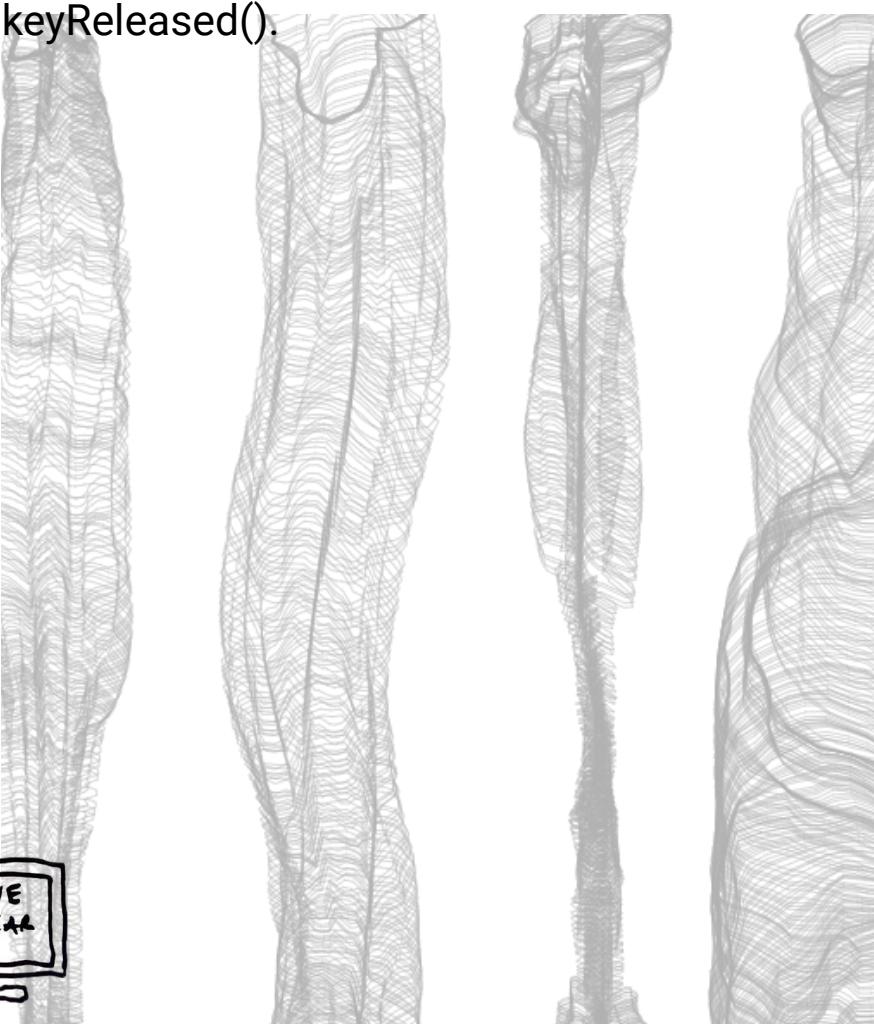


//GRAPHIX GEN 3000

flow charts



I used a function-based workflow to code my graphics. I tried to keep my code neat by wrapping each brush in its own function, and then called that function in draw() through adding numbered key bindings. Values were passed between global and local variables, preload() and setup(), to draw(), my custom brush functions and to keyPressed() and keyReleased().



SETUP () {

```

    [ ]
    START == 0
    DISPLAY TXT.
  
```

DRAW() {

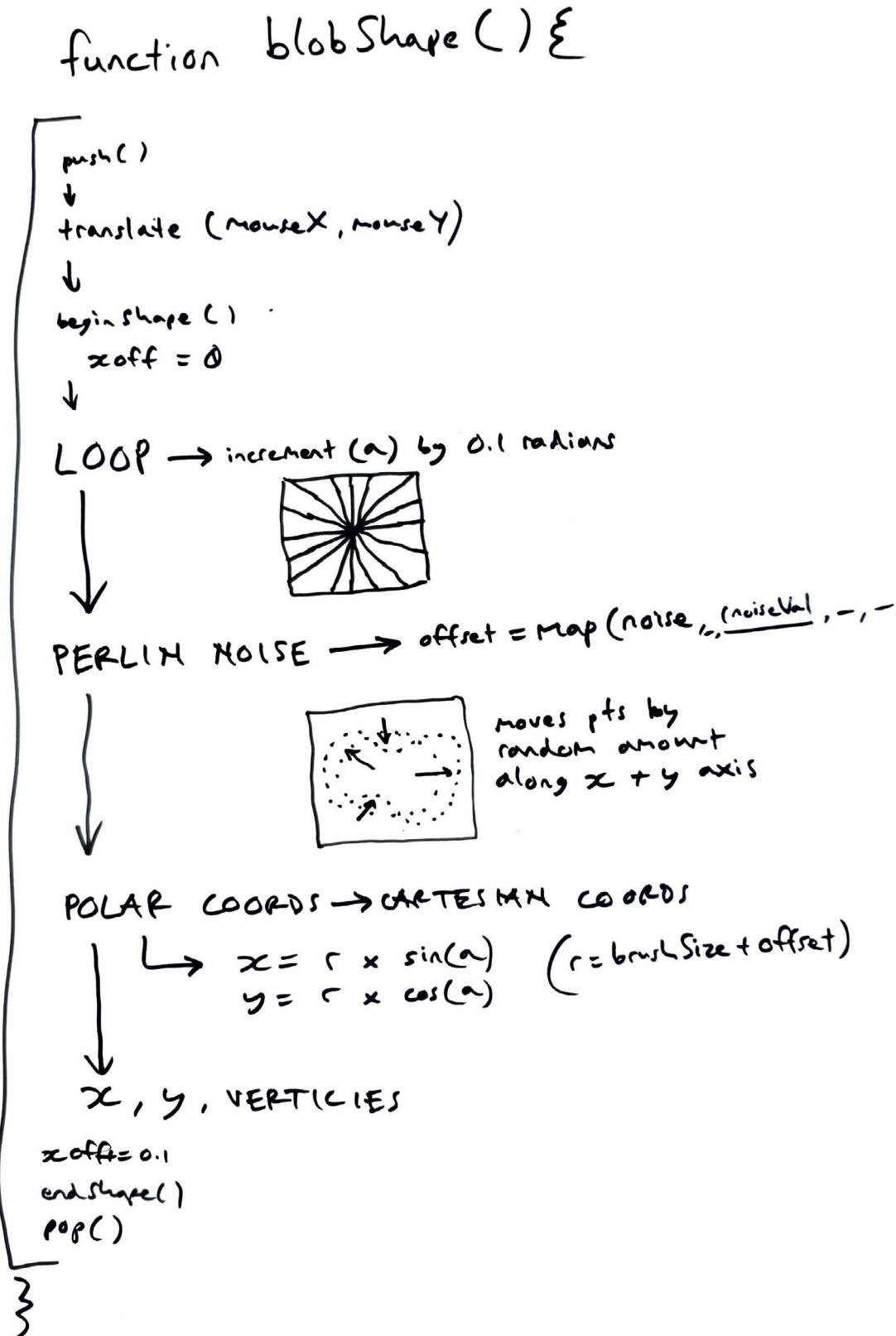
```

    [ ]
    START == 1 {
      [ ]
      DRAW MODE 1 ← Key == 1
      blobShape
      DRAW MODE 2 ← Key == 2
      blobDotShape
      DRAW MODE 3 ← Key == 3
      blobLineShape
      DRAW MODE 4 ← Key == 4
      manyRects
      DRAW MODE 5 ← Key == 5
      imgBrush
    }
  
```

To code the start screen I set an initial value (start == 0) in setup and wrote out my welcome text. Then when keyPressed == ENTER, the background was set to white and start == 1. This then allowed all the draw() code to be called in a single if(){} statement.

//GRAPHIX GEN 3000

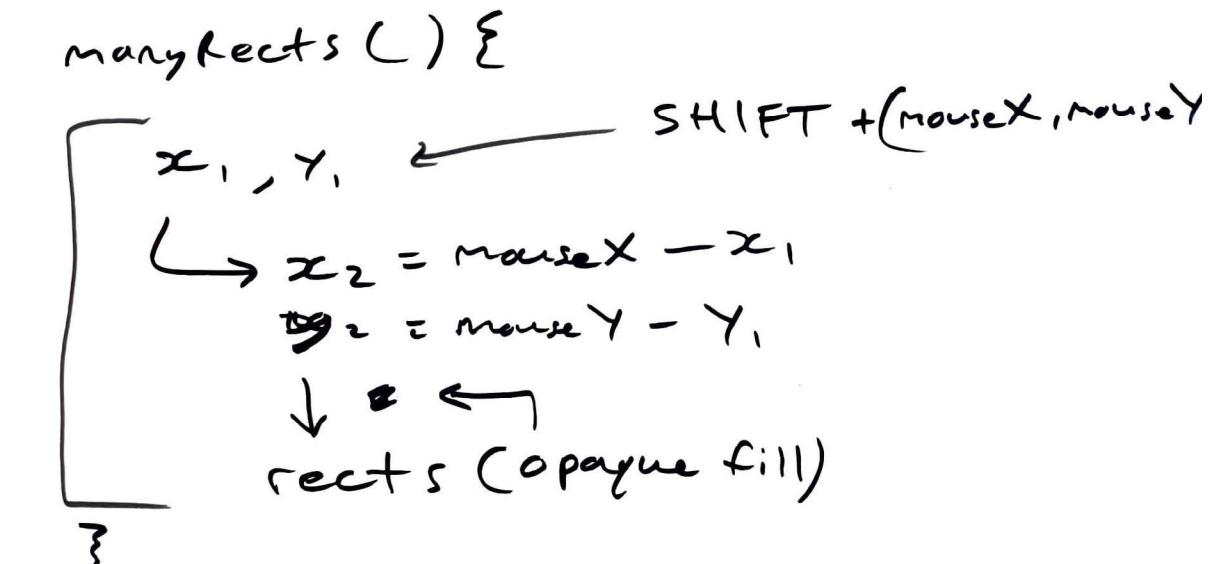
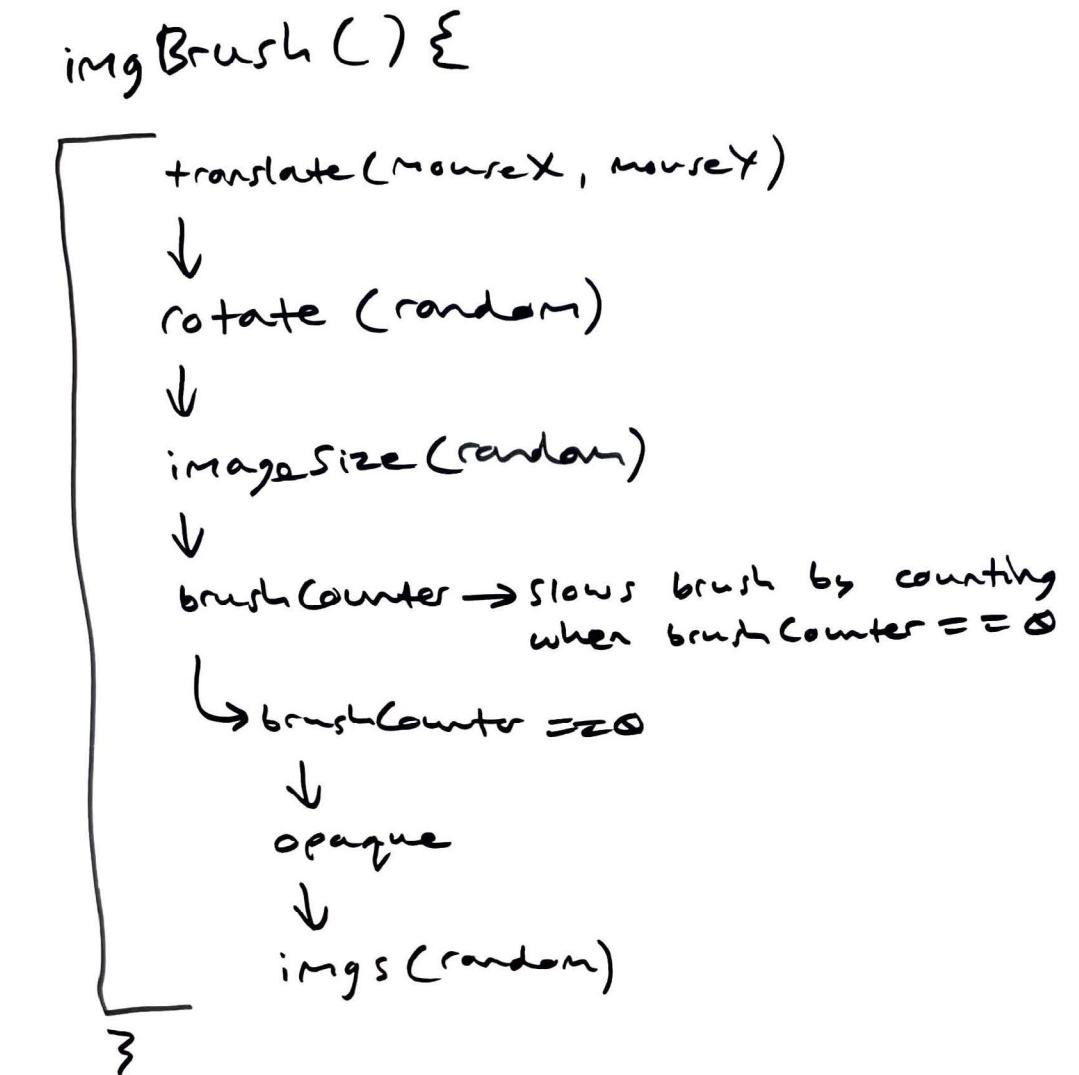
flow charts



My organic blob shape was made with help from the Coding Train. It creates a large number of vertices around a central point (`mouseX, mouseY`). These vertices move each frame using Perlin Noise along the x and y axis. The noise can be adjusted with key bindings.

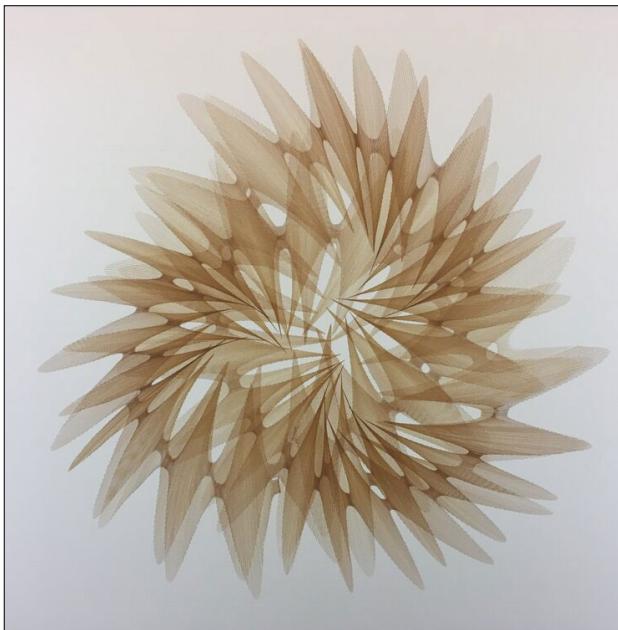
My image brush uses several preloaded() images. The brush scatters the images by random rotates, opacity adjustment, and size. It scatters slowly by using an increment counter that calls an image every 0.

Many rect sets and initial `x1` and `y1` value and colour through a keybind. The `x2` and `y2` values are calculated when the mouse is pressed or dragged.



//GRAPHIX GEN 3000

artistic influences

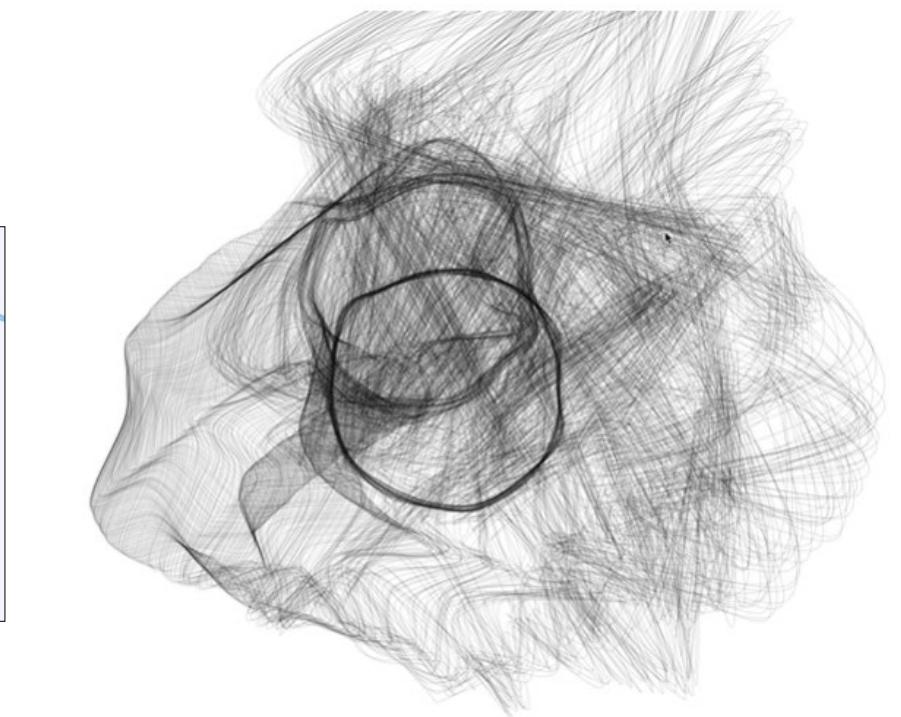
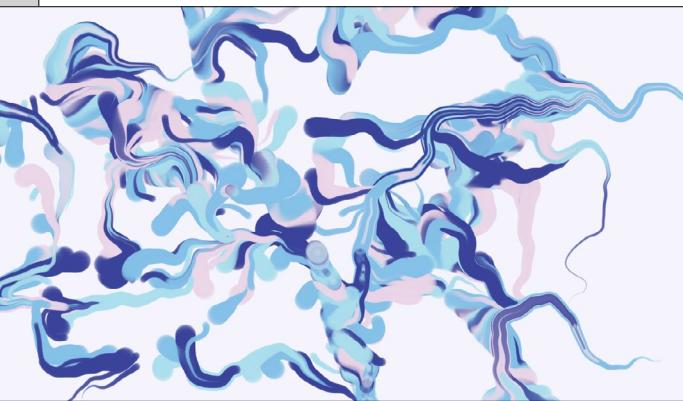
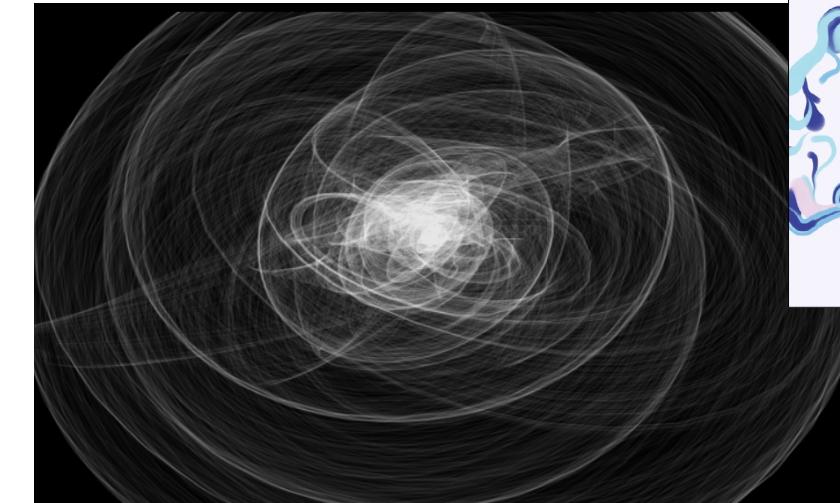
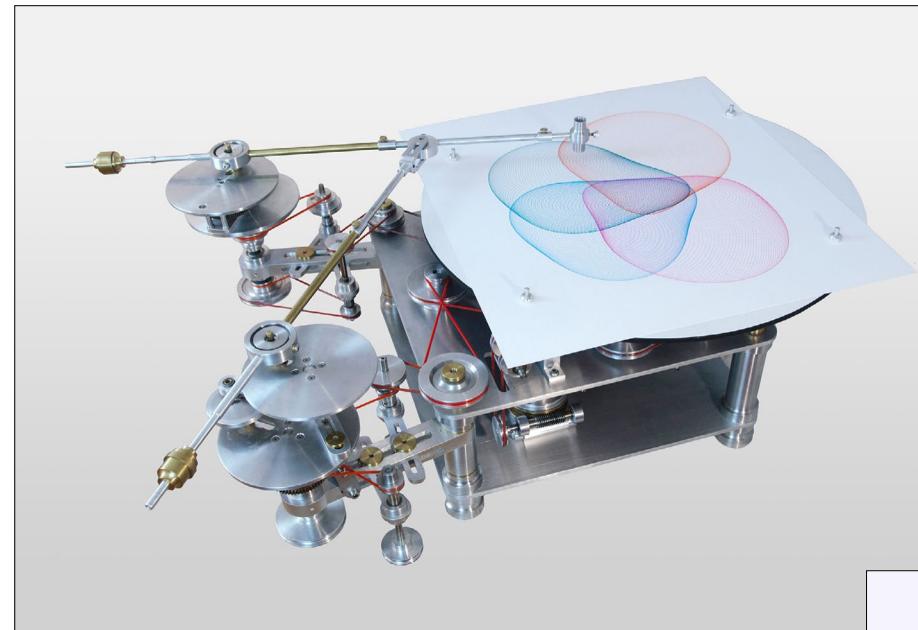


Gandyworks
<http://www.jamesnolangandy.com/>

Generative gestaltung
<http://www.generative-gestaltung.de/2/>
http://www.generative-gestaltung.de/1/P_2_2_3_02

Calligraphy
<https://openprocessing.org/sketch/1514071>
<https://openprocessing.org/sketch/793375>

Particles circulate
<https://openprocessing.org/sketch/386707>
Colour gen
<https://openprocessing.org/sketch/1179185>



Generative Design About Sketches Contact

DE / EN / JP

P_2_2_6_02 P_2_2_6_03 P_2_2_6_04 P_2_2_6_05 P_2_3_1_01

P_2_3_1_02 P_2_3_2_01

Code → P_2_2_3_02

//GRAPHIX GEN 3000

references

McLean, W. [Will Mclean]. (2020, March 31). P5.js - Image brush [Video].

YouTube. <https://www.youtube.com/watch?v=WhS1x9wSGC0>

Schiffman, D. (n.d.). The Coding Train. The Coding Train. Retrieved April 4, 2022, from <https://thecodingtrain.com/>

Schiffman, D. (2016). CC 36: Blobby (1.0.0) [Computer code]. The Coding Train. https://editor.p5js.org/codingtrain/sketches/L_ME8qKmQ

The Coding Train. (2016, September 2). Coding Challenge #36: Blobby! [Video]. YouTube. <https://www.youtube.com/watch?v=rX5p-QRP6R4>