Processing Community Day 2019
Brisbane

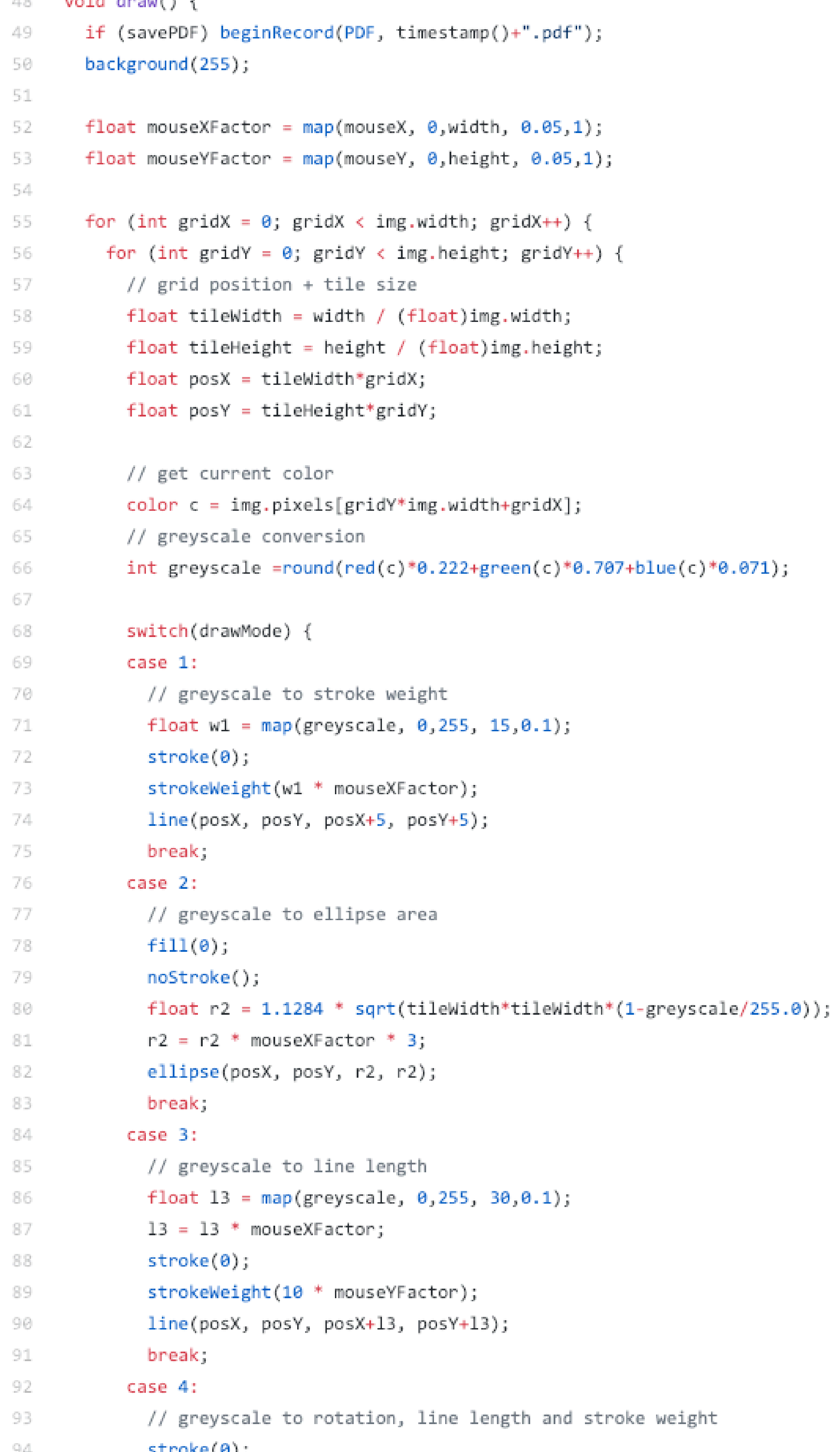
PCD

Griffith
UNIVERSITY
Queensland
College of Art

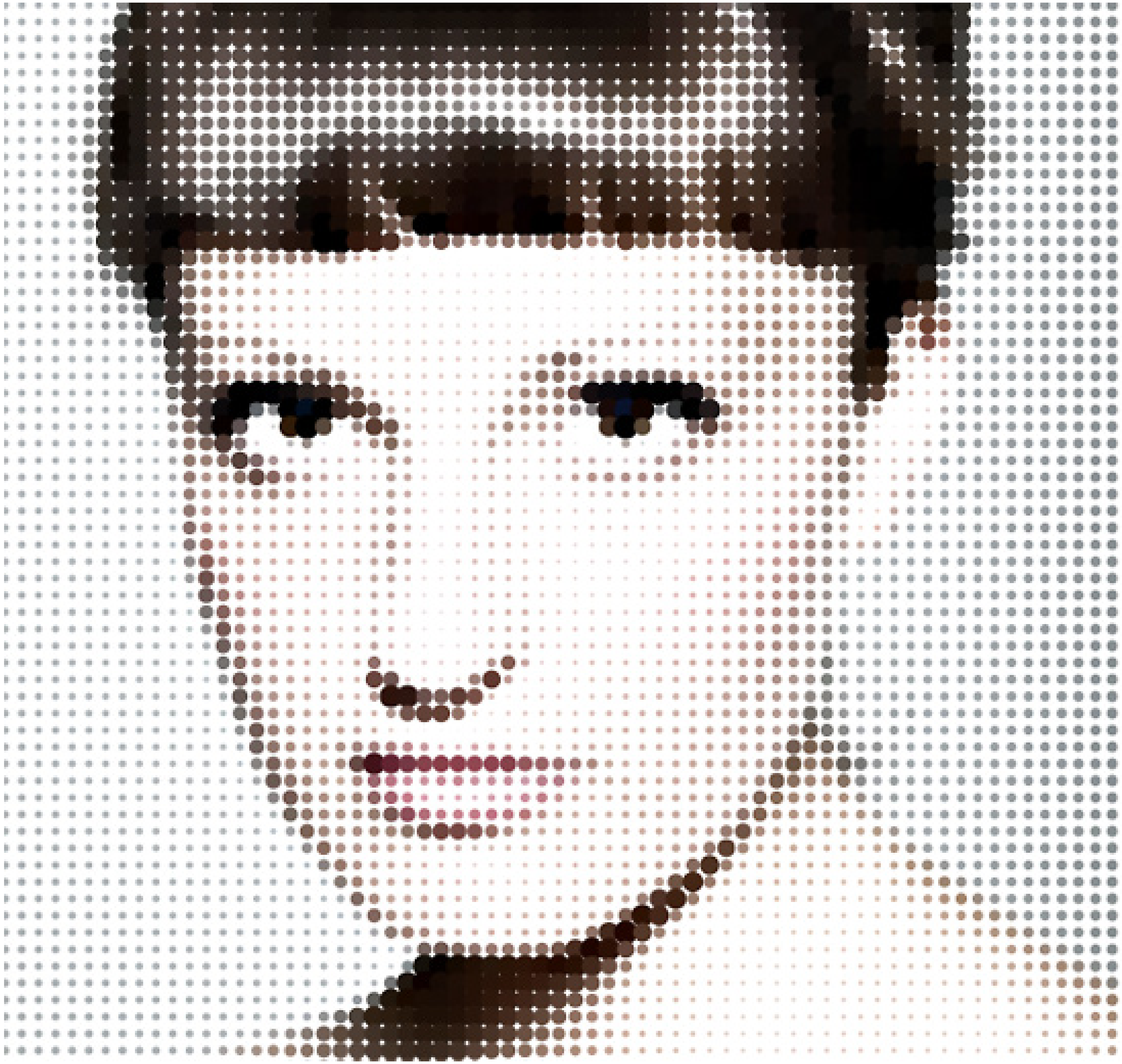
Queensland, Australia

Today's schedule

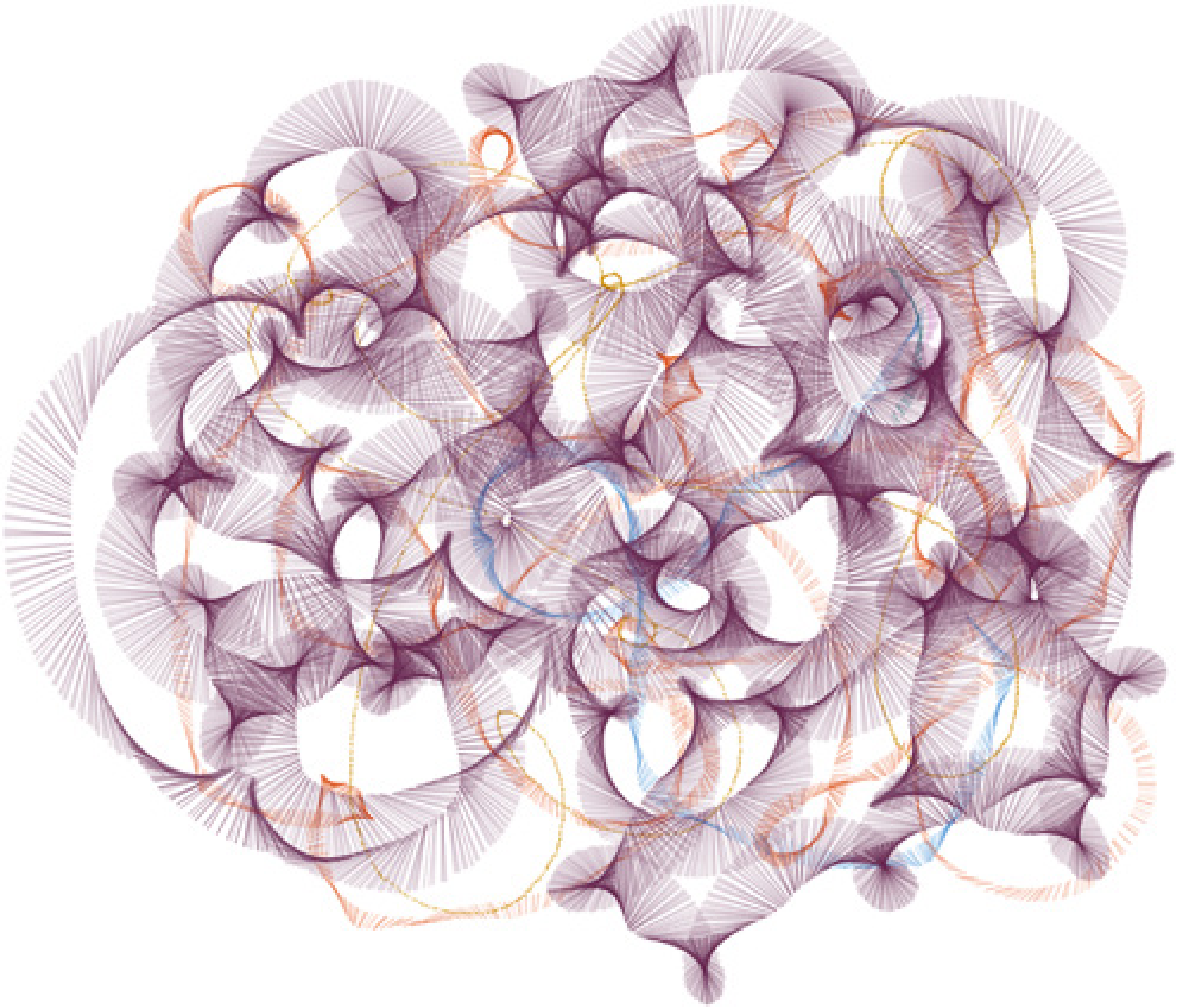
10.00 - 10.30am	Welcome Introductions and code jam briefing
10.30 - 11.00am	Beginner workshop/Experienced meetup Choose the group you feel most comfortable in
11.00 - 11.30am	Live stream showcase Brisbane, Melbourne, and Wellington
11.30 - 1.30pm	Open session Code jam, beginner assistance, advanced dev planning, share demos, something else?
1.30 - 2.00pm	Open mic The stage is yours to show something you've made



```
void draw() {
  if (savePDF) beginRecord(PDF, timestamp()+".pdf");
  background(255);

  float mouseXFactor = map(mouseX, 0,width, 0.05,1);
  float mouseYFactor = map(mouseY, 0,height, 0.05,1);

  for (int gridX = 0; gridX < img.width; gridX++) {
    for (int gridY = 0; gridY < img.height; gridY++) {
      // grid position + tile size
      float tileWidth = width / (float)img.width;
      float tileHeight = height / (float)img.height;
      float posX = tileWidth*gridX;
      float posY = tileHeight*gridY;

      // get current color
      color c = img.pixels[gridY*img.width+gridX];
      // greyscale conversion
      int greyscale =round(red(c)*0.222+green(c)*0.707+blue(c)*0.071);

      switch(drawMode) {
      case 1:
        // greyscale to stroke weight
        float w1 = map(greyscale, 0,255, 15,0.1);
        stroke(0);
        strokeWeight(w1 * mouseXFactor);
        line(posX, posY, posX+5, posY+5);
        break;
      case 2:
        // greyscale to ellipse area
        fill(0);
        noStroke();
        float r2 = 1.1284 * sqrt(tileWidth*tileWidth*(1-greyscale/255.0));
        r2 = r2 * mouseXFactor * 3;
        ellipse(posX, posY, r2, r2);
        break;
      case 3:
        // greyscale to line length
        float l3 = map(greyscale, 0,255, 30,0.1);
        l3 = l3 * mouseXFactor;
        stroke(0);
        strokeWeight(10 * mouseYFactor);
        line(posX, posY, posX+l3, posY+l3);
        break;
      case 4:
        // greyscale to rotation, line length and stroke weight
        stroke(0);
```

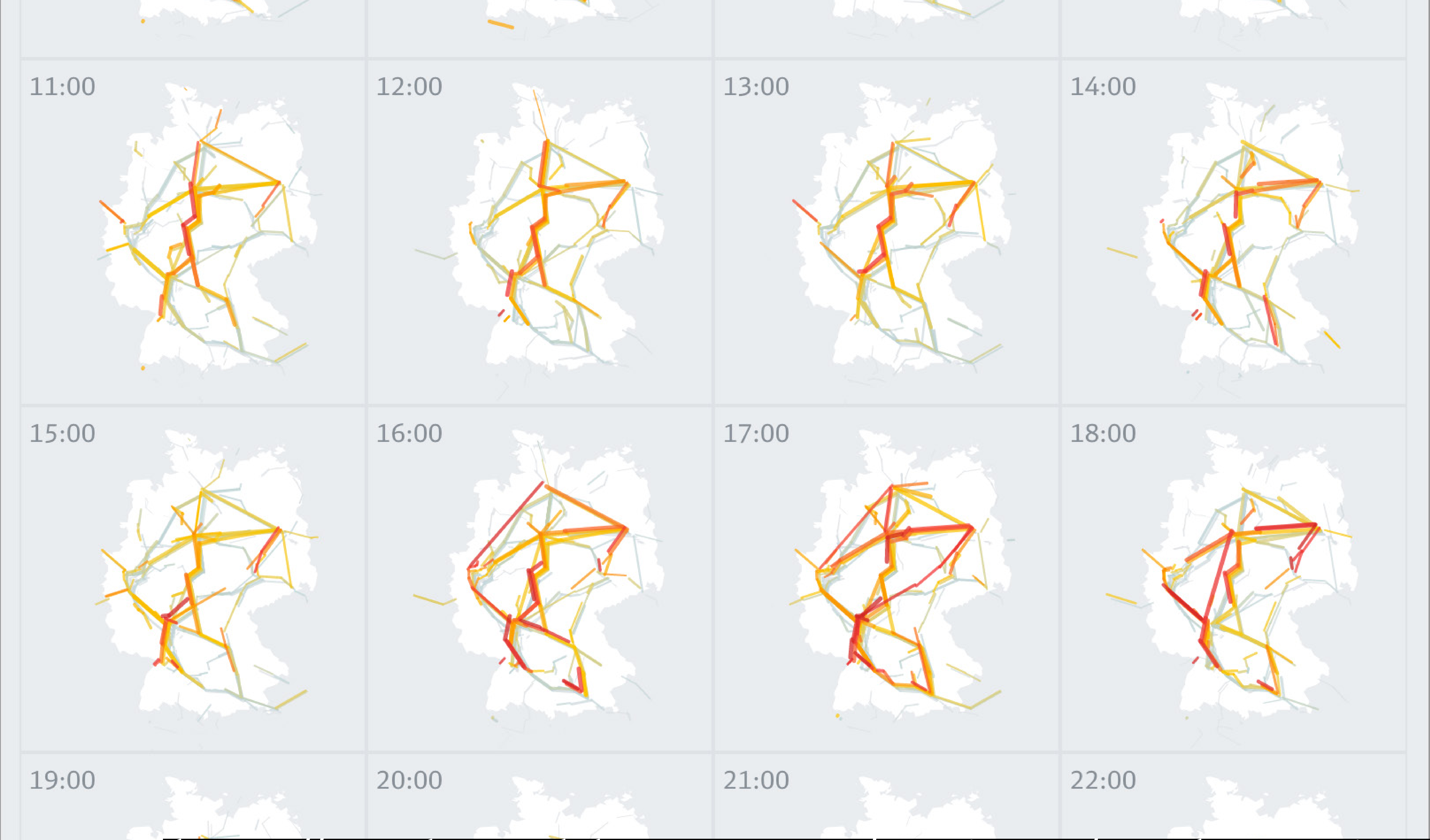


```
color col = color(181,157,0,100);
float lineModuleSize = 0;
float angle = 0;
float angleSpeed = 1.0;
PShape lineModule = null;

int clickPosX = 0;
int clickPosY = 0;


void setup() {
  // use full screen size
  size(displayWidth, displayHeight);
  background(255);
  smooth();
  cursor(CROSS);
}

void draw() {
  if (mousePressed) {
    int x = mouseX;
    int y = mouseY;
    if (keyPressed && keyCode == SHIFT) {
      if (abs(clickPosX-x) > abs(clickPosY-y)) y = clickPosY;
      else x = clickPosX;
    }

    strokeWeight(0.75);
    noFill();
    stroke(col);
    pushMatrix();
    translate(x, y);
    rotate(radians(angle));
    if (lineModule != null) {
      shape(lineModule, 0, 0, lineModuleSize, lineModuleSize);
    }
    else {
      line(0, 0, lineModuleSize, lineModuleSize);
    }
    angle = angle + angleSpeed;
    popMatrix();
  }
```

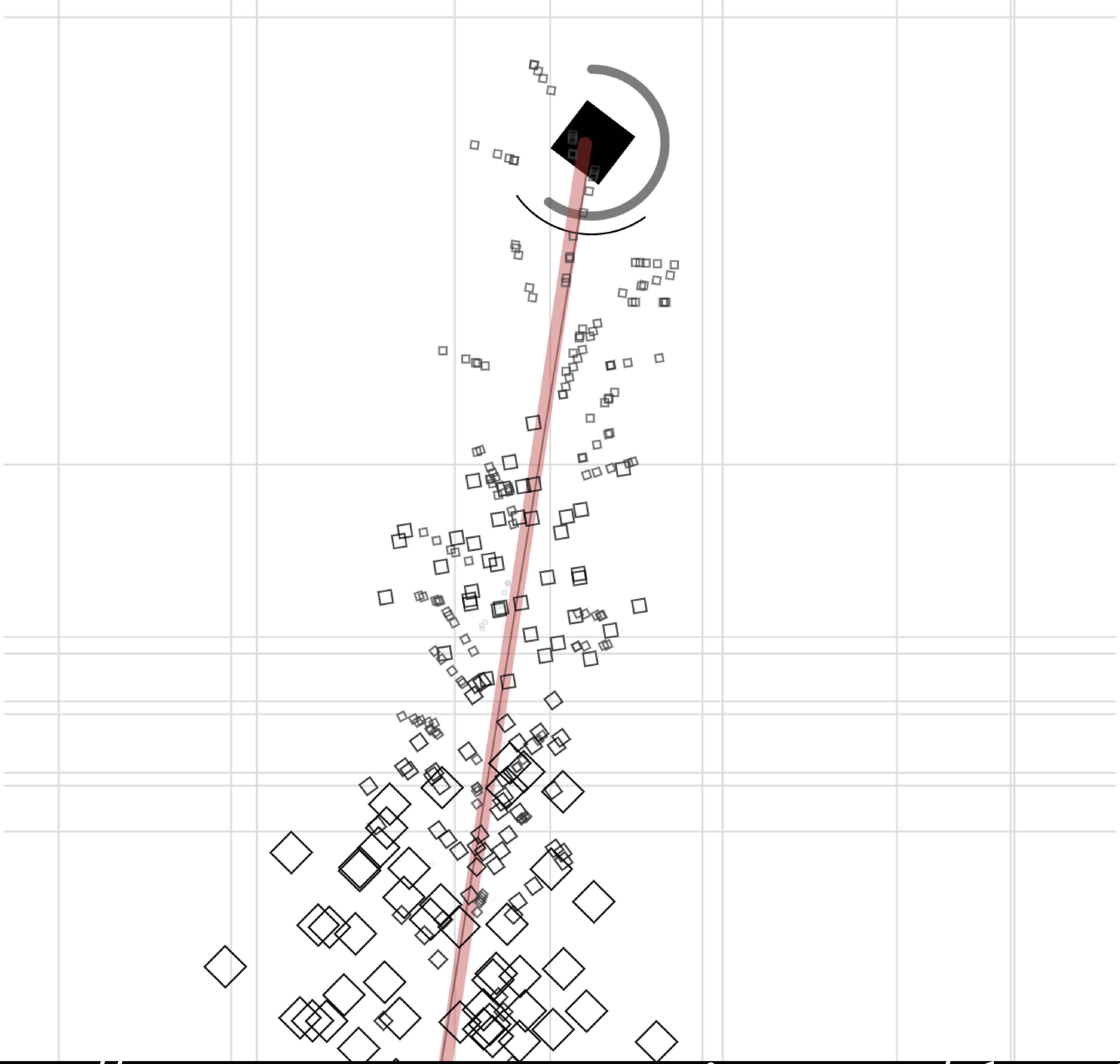


http://www.sanchtv.com/?p=81

http://studioanf.com/v0id-6/



http://truth-and-beauty.net/projects/peakspotting



https://www.openprocessing.org/sketch/453716

Who are we all?

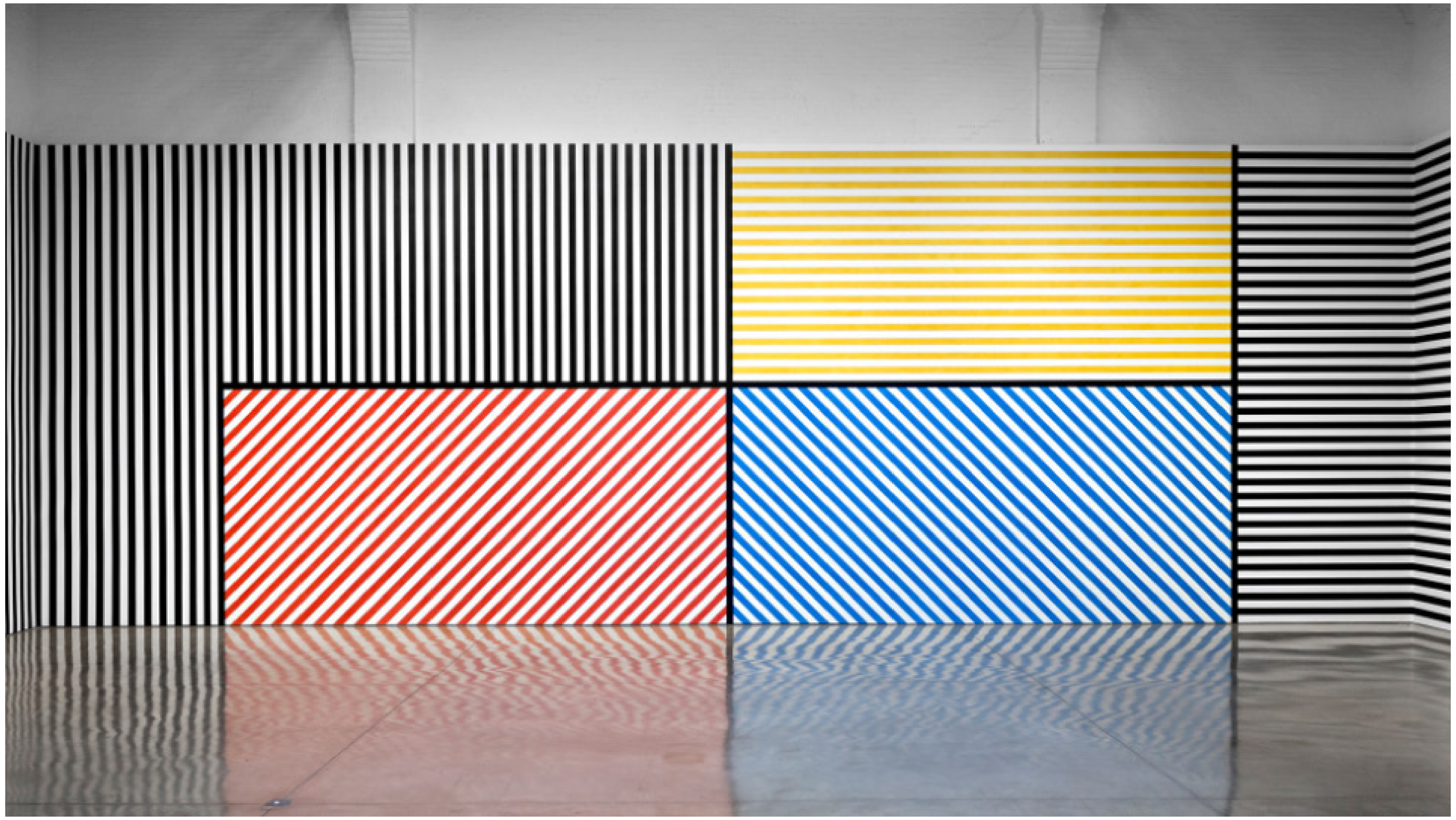
In only 1-3 words for each of these, tell us:

Your name

Beginner/intermediate/advanced

What job you’d have on
a pirate ship

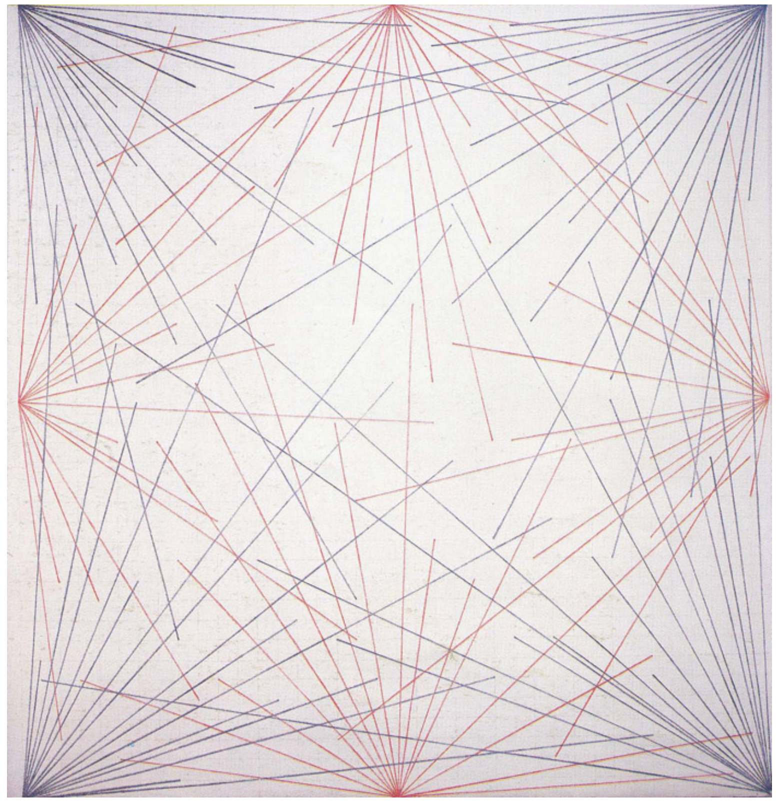
Processing Sol



Sol LeWitt was an American conceptual and minimalist artist. His best known works are his “Wall Drawing” series of nearly 1300 works created between 1968 and his death in 2007.

Each “work” is a set of instructions for a professional draftsperson to execute on a wall, sometimes with latitude for interpretation.

Our challenge today is to execute his wall drawings on the canvas of a screen.



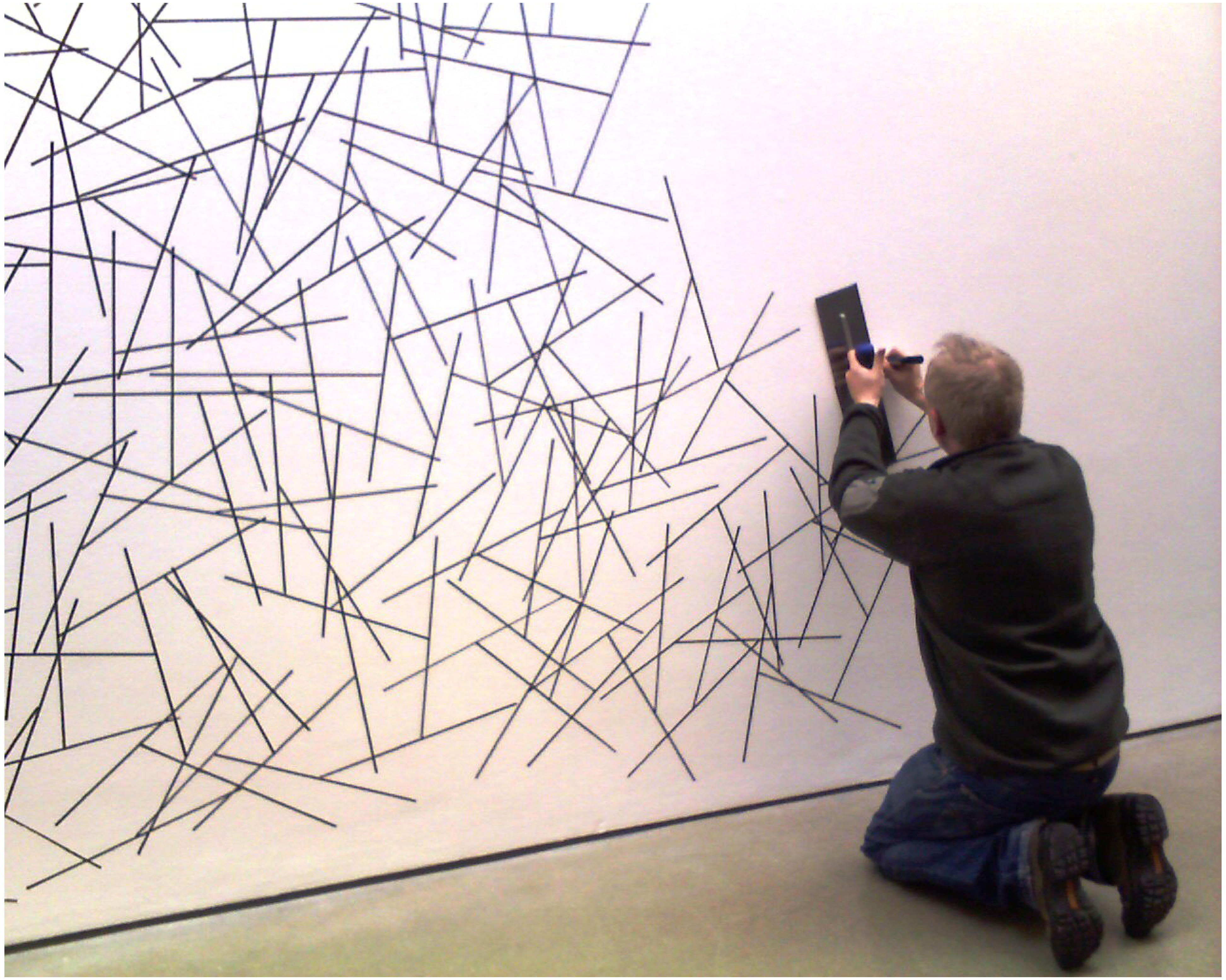
Wall Drawing #46 (1970)

Vertical lines,
not straight, not
touching, uniformly
dispersed with
maximum density,
covering the entire
surface of the wall.

PCD

Wall Drawing #86 (1971)

Ten thousand lines about 10 inches (25 cm) long, covering the wall evenly.



PCD

PCD Code Challenge

Write Processing or p5.js code to execute the instructions of a Sol LeWitt wall drawing, interpreted for screen.

Output the final image the code creates or take a screenshot. The solutions will be collected and printed in a zine.

Submit your image along with your name, wall drawing number, and a link to your code (optional) through the form here: http://processingday2019.cmp.ac.nz/

PCD Advanced Code Challenge

Write Processing or p5.js code to write sets of instructions in the style of Sol LeWitt and then execute those instructions. (Interpret this challenge as you see fit.)

This project is inspired by Solving Sol: http://solvingsol.com/