

# Recoding project

8 *Ecke* de Georg Nees

# Artwork chosen

Title : 8-ecke

Artist : Georg Nees

Year : 1964

Technique : ink on paper, plotter drawing

Computer : Siemens-Digitalrechner 2002

Software : ALGOL

Plotter : ZUSE-Graphomat Z 64 plotter

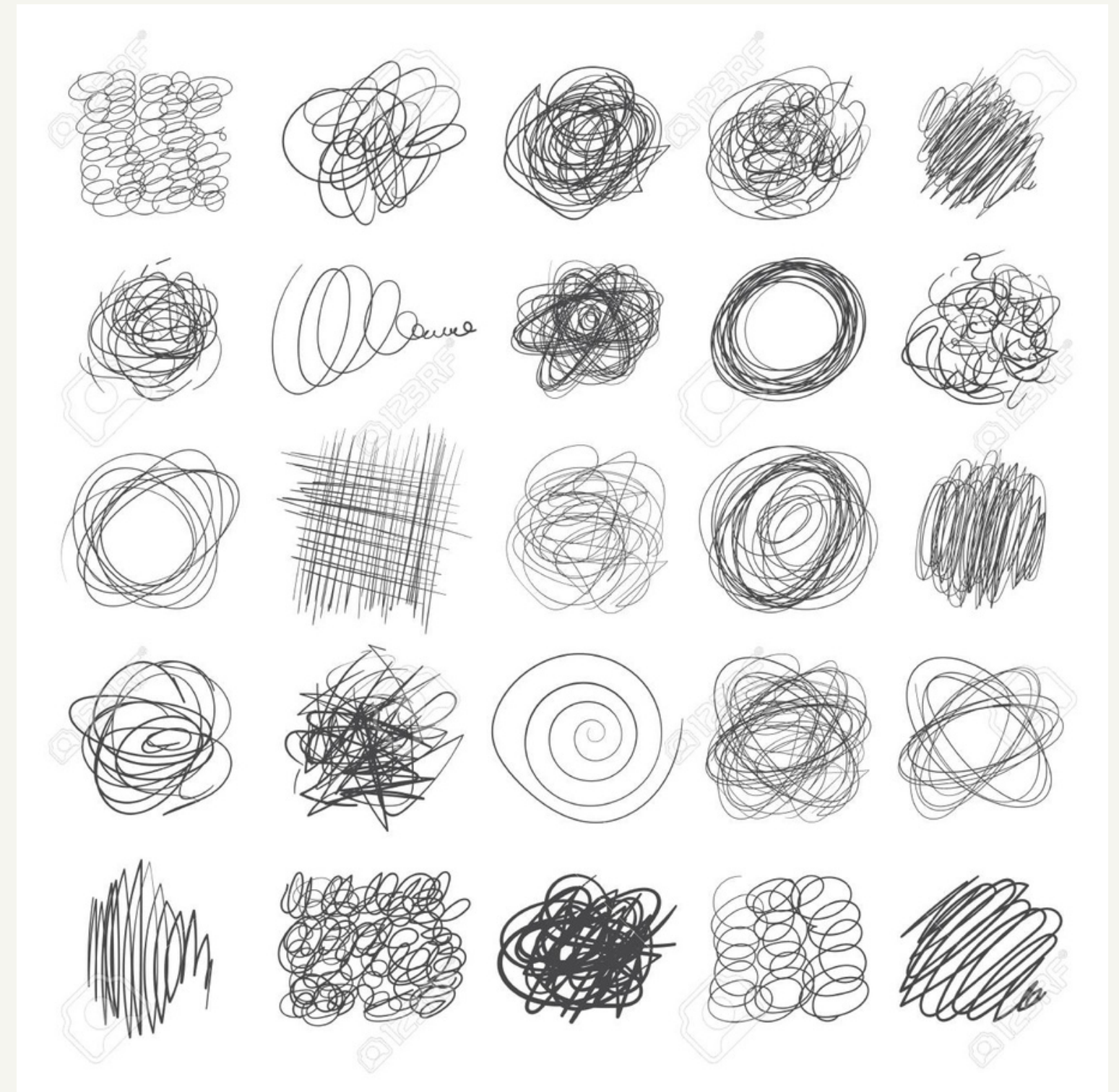
Dimensions : 29,7 x 41,8



# Reasons and description

I choose this artwork because it reminds me of the shape I use to draw when I'm on the phone with someone. It's kind of automatic drawing and I really love the shape it takes.

It's made of 266 polygons, each of them have 8 vertices. They are disposed equally on the surface of the canvas, without overlapping.



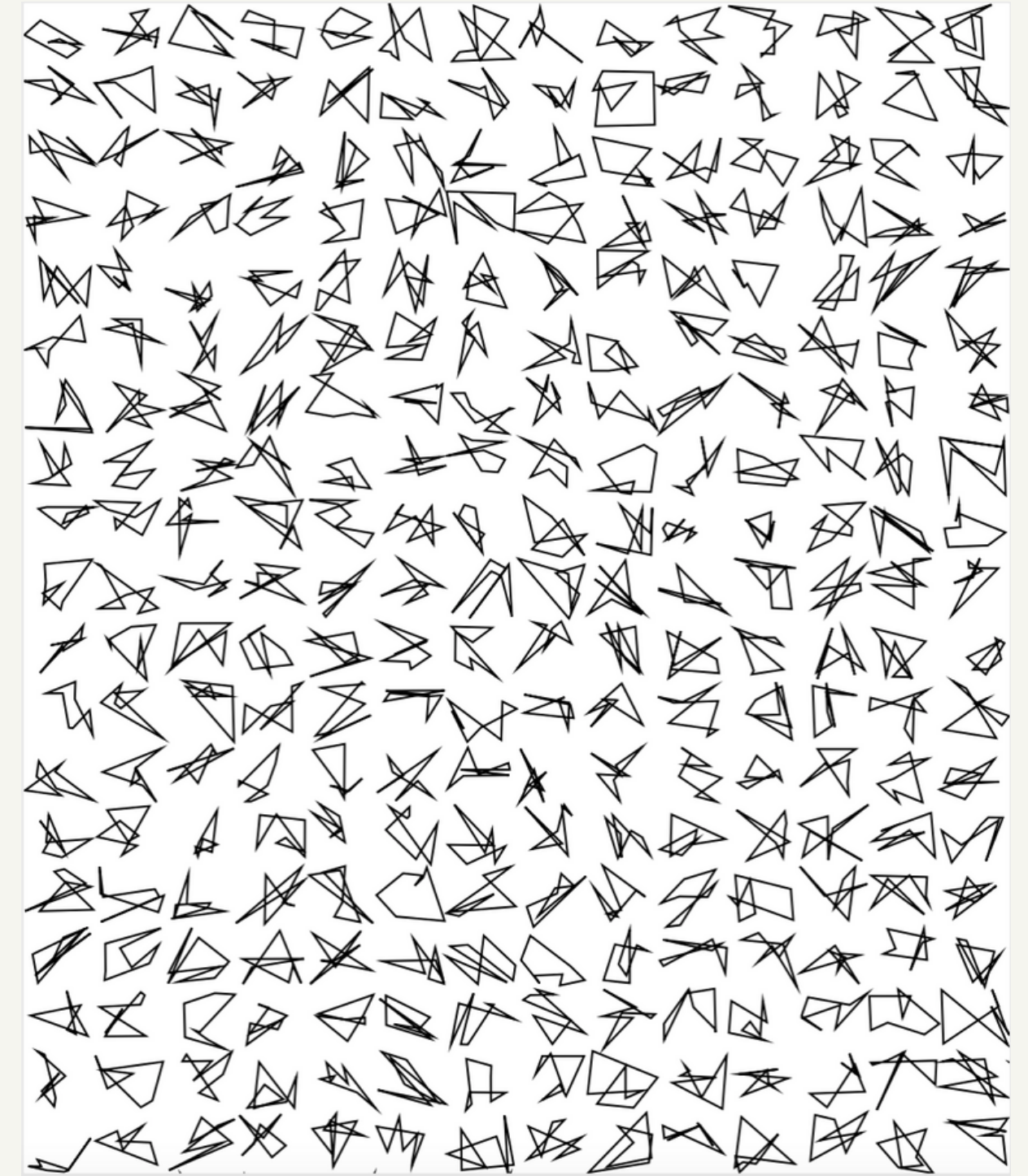
# Pseudo code and first version

Divide your canvas with 19 rows and 14 columns then, on each case you'll draw a point with random coordinates. Then draw a second one and connect them both with a line. Repeat this operation until you have 8 vertices then connect the last one with the first.

```
//first code
for(let x=1; x<(params.nb_Columns+2); x++){
  for(let y=1; y<(params.nb_Lines+2); y++){
    for(let k=0; k<8; k++){
      //picking a random point
      let X=random((x-1)*(width/params.nb_Columns),(x)*(width/params.nb_Columns))
      let Y=random((y-1)*(height/params.nb_Lines),(y)*(height/params.nb_Lines))
      if(k==0){
        premierX=X
        premierY=Y
        precedentX=premierX
        precedentY=premierY
      }else{
        //Draw a line between points
        line(X, Y, precedentX, precedentY)
        //Closing the shape
        if(k==7){
          line(X, Y, premierX, premierY)
        }
        precedentX=X;
        precedentY=Y;
      }
    }
  }
}
```

# Explication and rendering

I first didn't know how to do it properly with P5 so I made it by my own, picking a point, draw a line between him and the previous one. It always needs to remember where the previous point is. It works well but the code could be smaller. It has the same kind of rendering as the original.



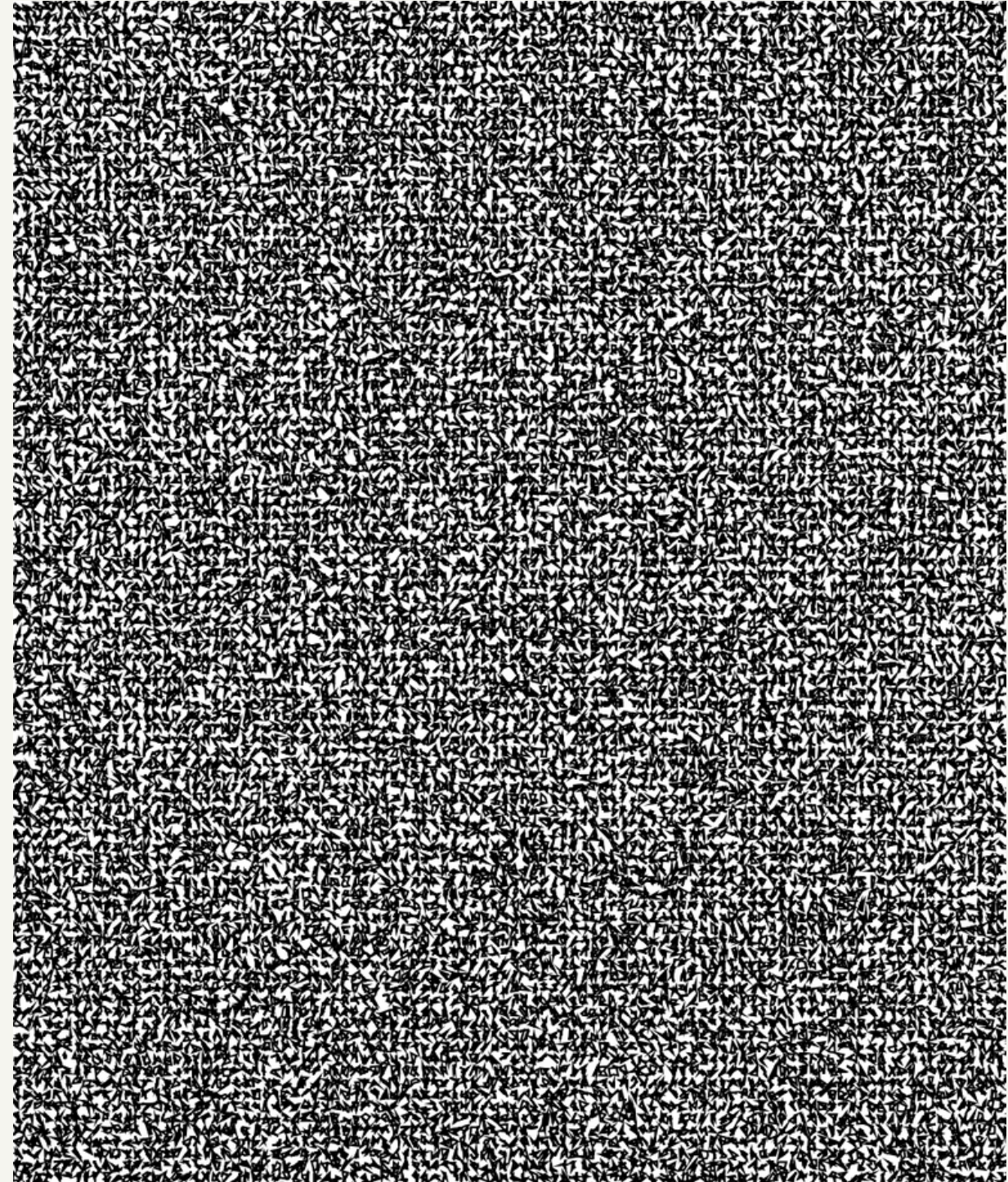
# Second version

Then, Jules told me there is a way to do it with vertex in P5. It has the same rendering but it's another way to write the code and it can be easier to understand when you know P5.

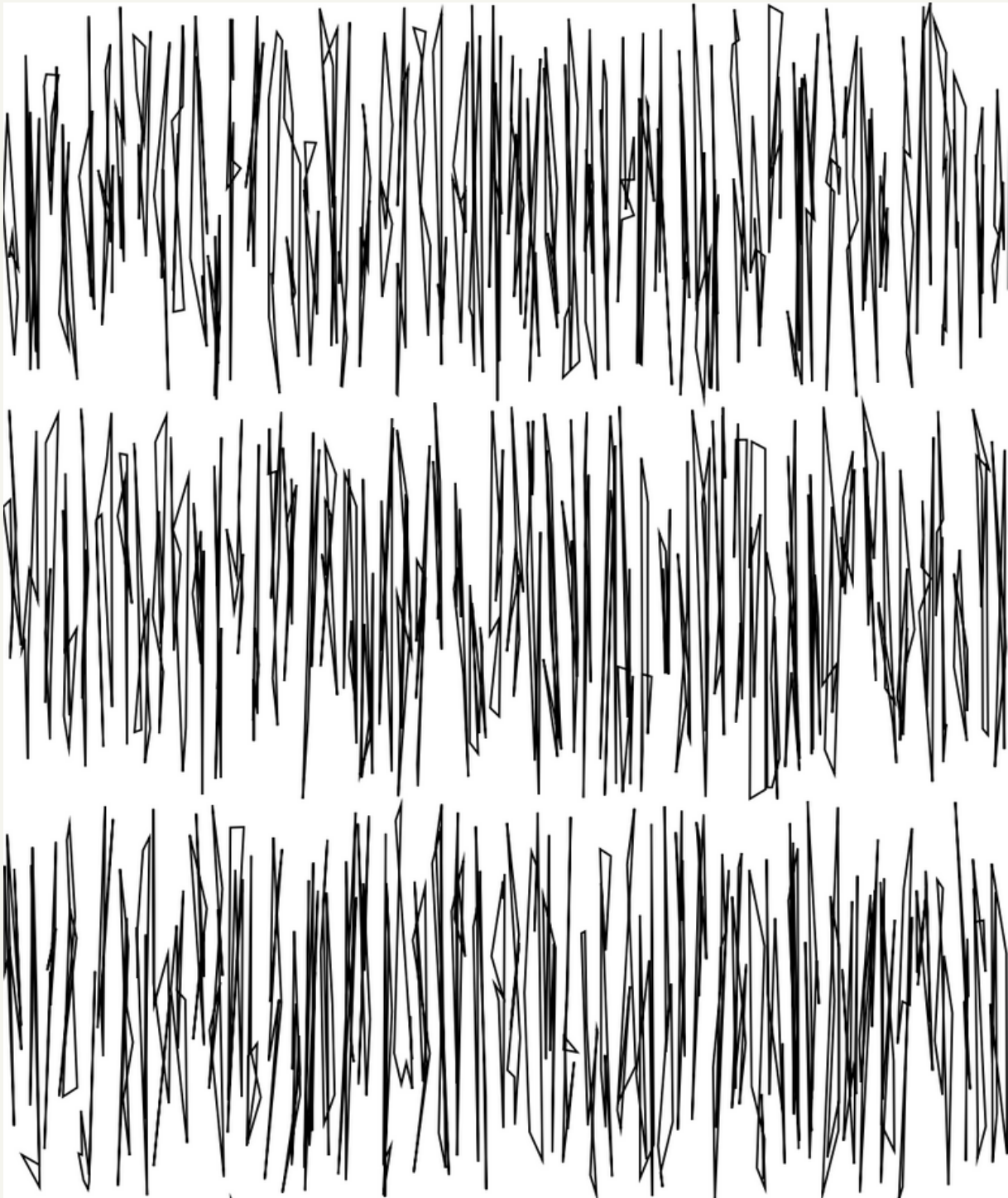
```
//Second code
noFill()
for(let x=1; x<(params.nb_Columns+2); x++){
  for(let y=1; y<(params.nb_Lines+2); y++){
    beginShape()
    for(let i=0; i<8; i++){
      //Picking a point
      let X=random((x-1)*(width/params.nb_Columns), (x)*(width/params.nb_Columns))
      let Y=random((y-1)*(height/params.nb_Lines), (y)*(height/params.nb_Lines))
      //Draw a line between point
      vertex(X, Y)
    }
    //Closing shape
    endShape(CLOSE)
  }
}
```

# Interactivity

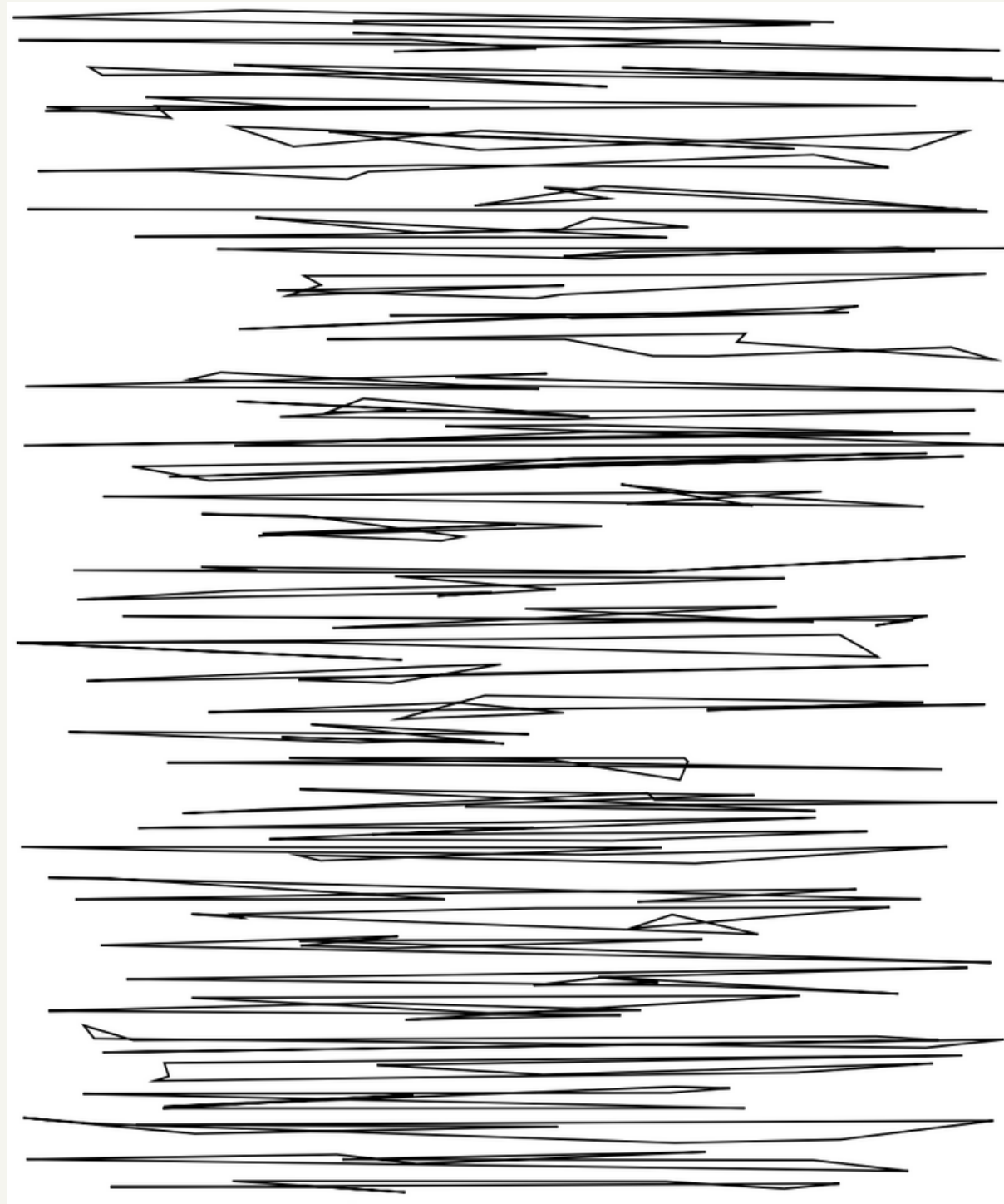
I decided to let user play a little bit with the code by offering him possibility to add some columns, lines or to change the random seed. It can give some interesting things like this "codex page"



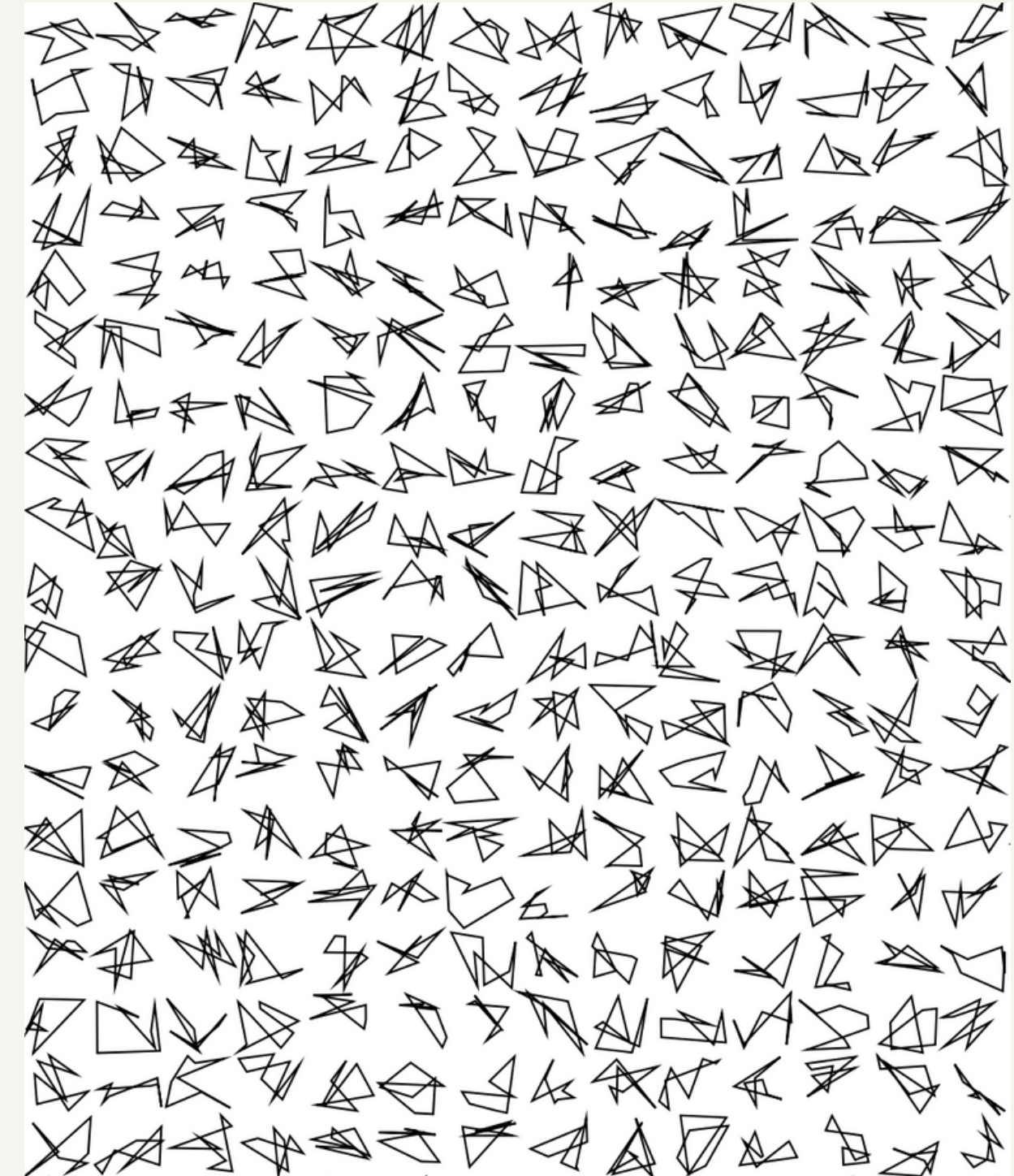
# Examples



lot of columns with few lines



lot of lines with few columns



standard with another random  
seed



# Presentation

Finally, when the user comes on the project, he first sees a presenting page to let him know about the recoding project and the original artwork. Then, he just has to click on the artwork to see my recoding

## Recoding project

For this project, i had to recode an artwork from a pioneer artist from the 60's, 70's.

I choose an artwork made by Georg Nees : "8-Ecke" that you can see below



Click on the image to see my recoding of this work

# Links

You can find my code on GitHub : [https://github.com/badbois/Recoding\\_Project](https://github.com/badbois/Recoding_Project)

And you can try it here : [https://badbois.github.io/Recoding\\_Project/](https://badbois.github.io/Recoding_Project/)