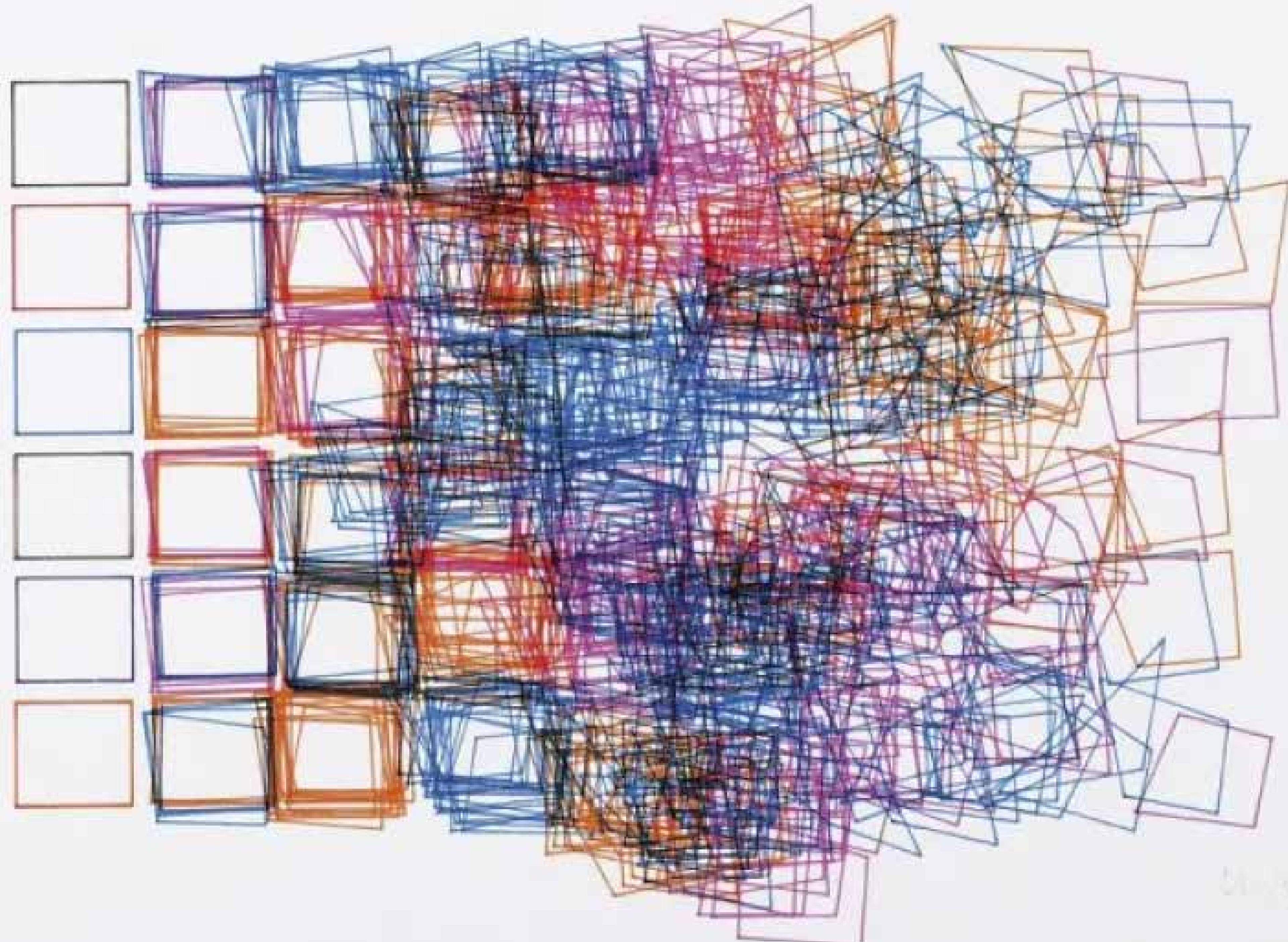
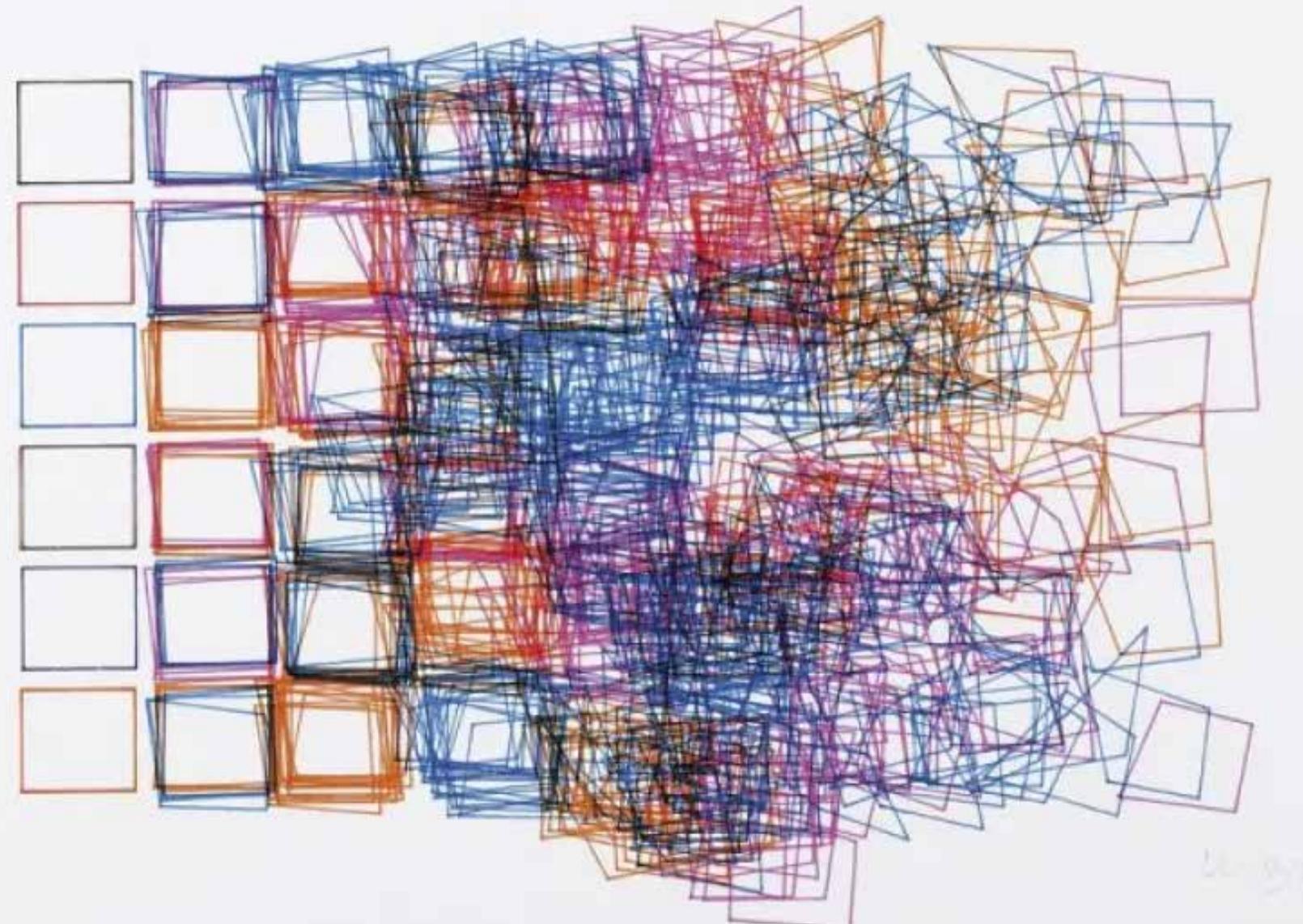
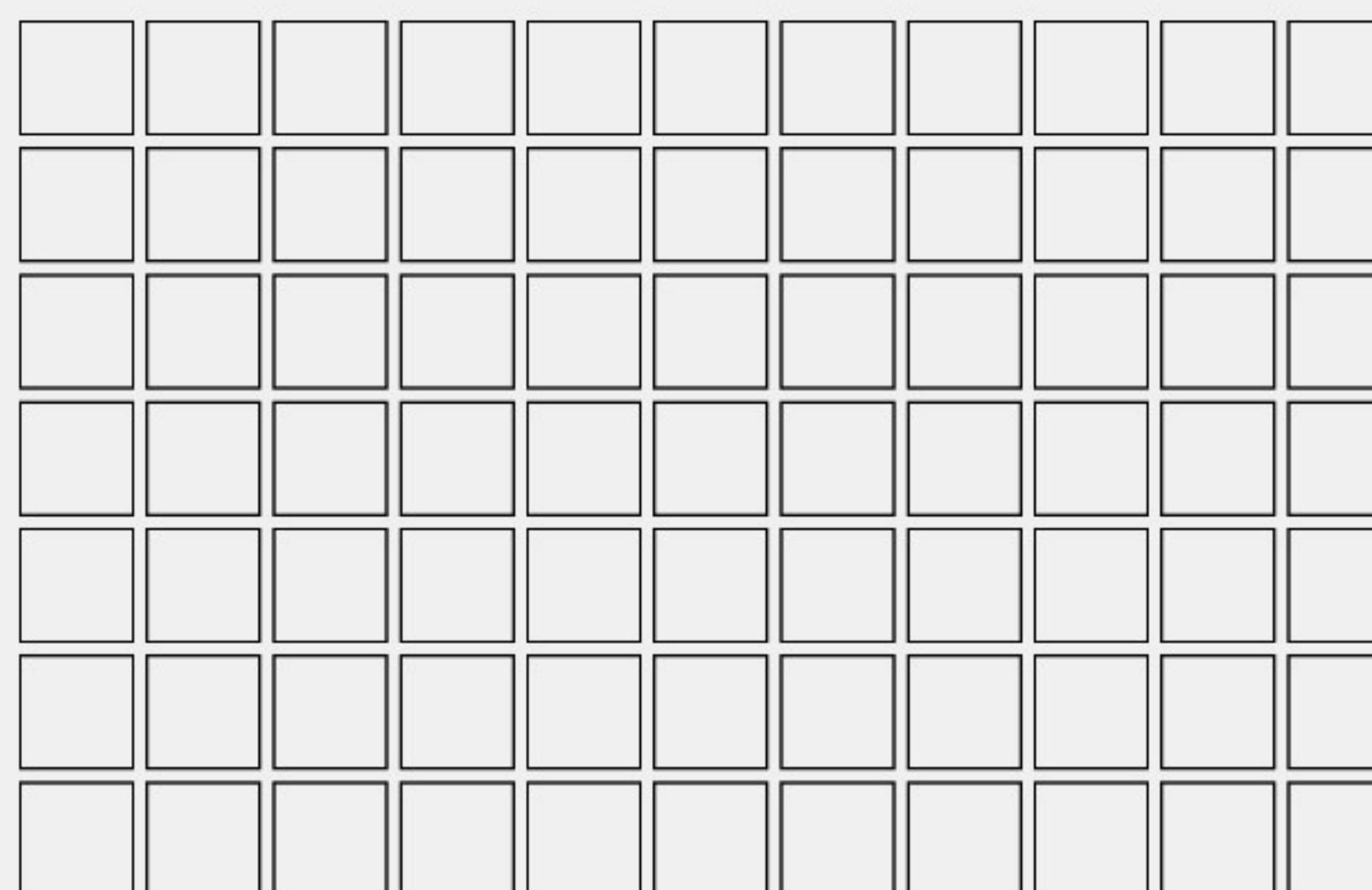


Sprint 1

Vera Molnár, “Structure de Quadrillatères”





Step 1 Creation of the grid with stacked squares. (soon to be changed into rectangles)

```
let columns = 11;
let rows = 7;
let space = 45;
let edge = 40;
let repetitions = 10;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * space + 50;
      let startY = r * space + 50;

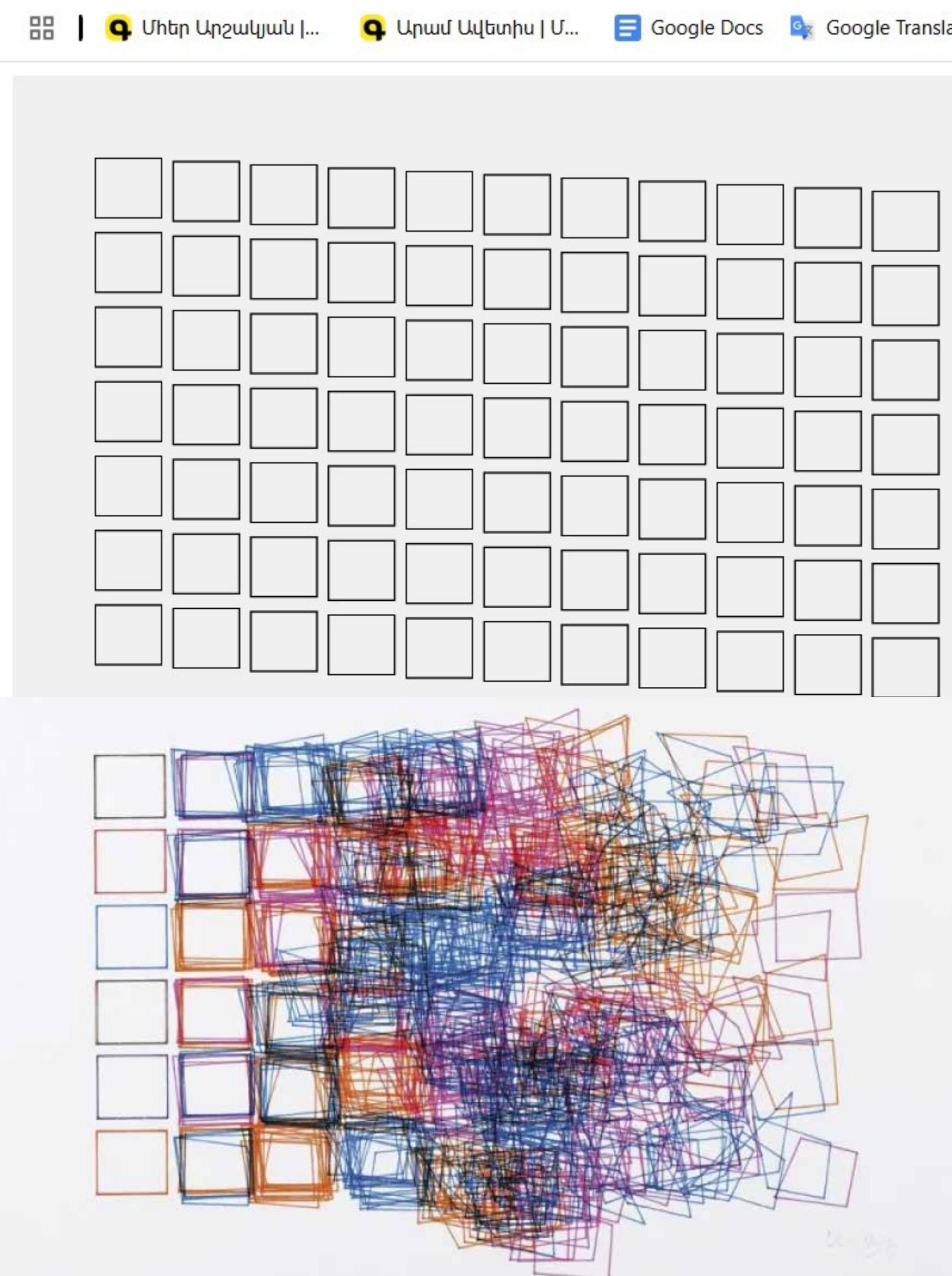
      for (let i = 0; i < repetitions; i++) {
        let xl = startX;
        let yl = startY;

        let x2 = startX + edge;
        let y2 = startY;

        let x3 = startX + edge;
        let y3 = startY + edge;

        let x4 = startX;
        let y4 = startY + edge;

        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```



Step 2 Trying to understand how to incorporate map() Function to increase distortion from first to last row

```
let distort = map(c, 0, columns - 1, 0, maxDistort);
//now i need to add the distort to the coordinates of the angles
for (let r = 0; r < rows; r++) {
  for (let c = 0; c < columns; c++) {
    let startX = c * space + 50;
    let startY = r * space + 50;

    let distort = map(c, 0, columns - 1, 0, maxDistort);

    for (let i = 0; i < repetitions; i++) {
      let xl = startX+distort;
      let yl = startY+distort;

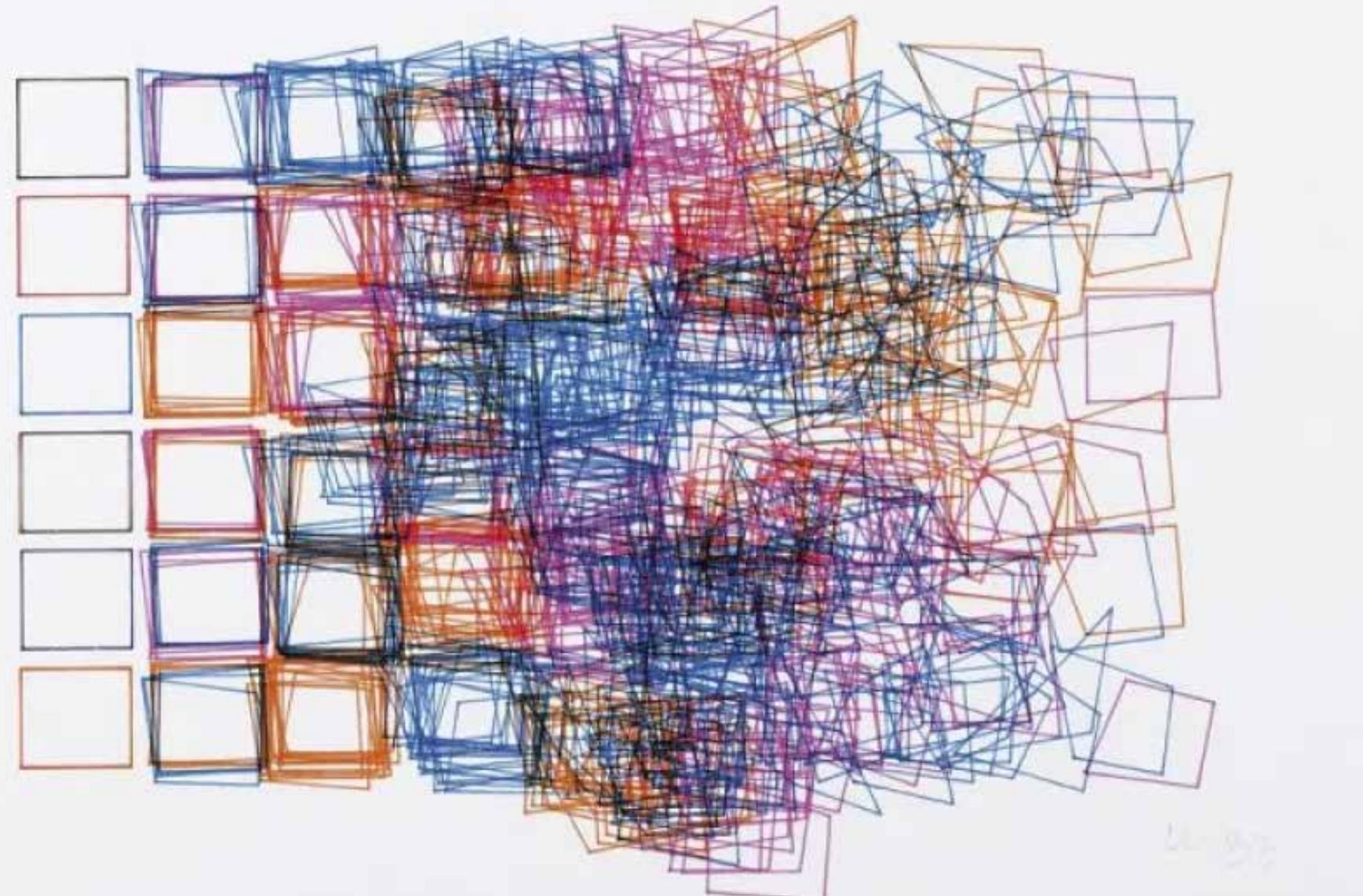
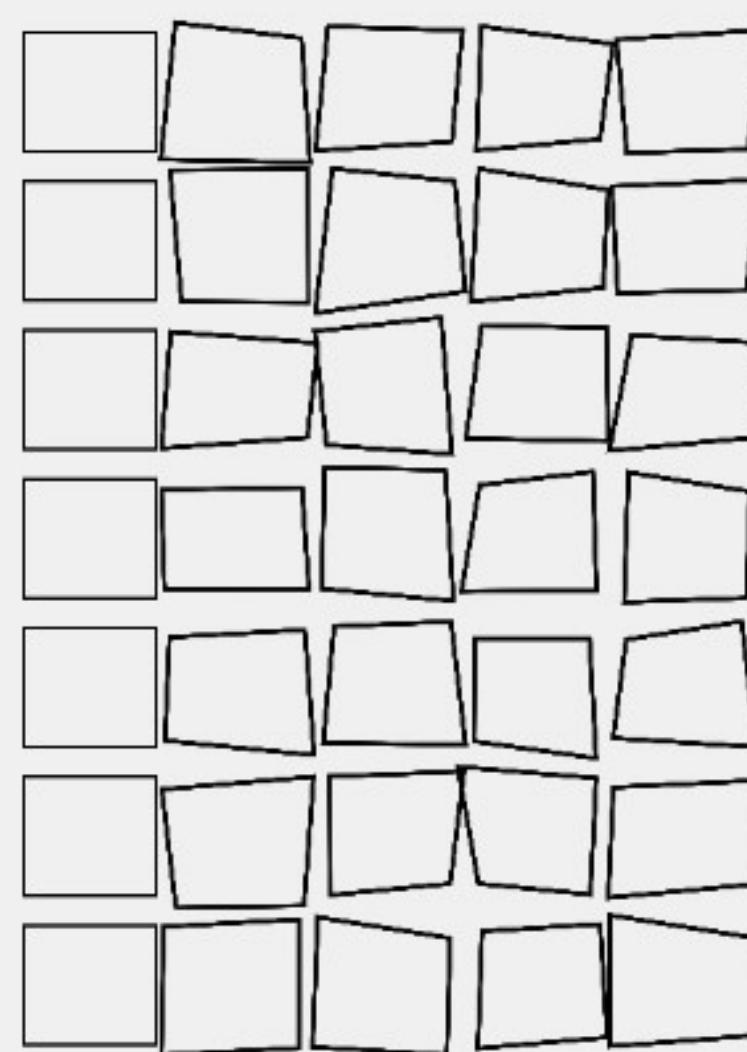
      let x2 = startX + edgeX+distort;
      let y2 = startY+distort;

      let x3 = startX + edgeX+distort;
      let y3 = startY + edgeY+distort;

      let x4 = startX+distort;
      let y4 = startY + edgeY+distort;

      quad(xl, yl, x2, y2, x3, y3, x4, y4);
    }
  }
}

Oops wrong. now they just moved diagonally. i need to add randomness? how do i generate randomness.
```



Step 3 Trying to add randomness

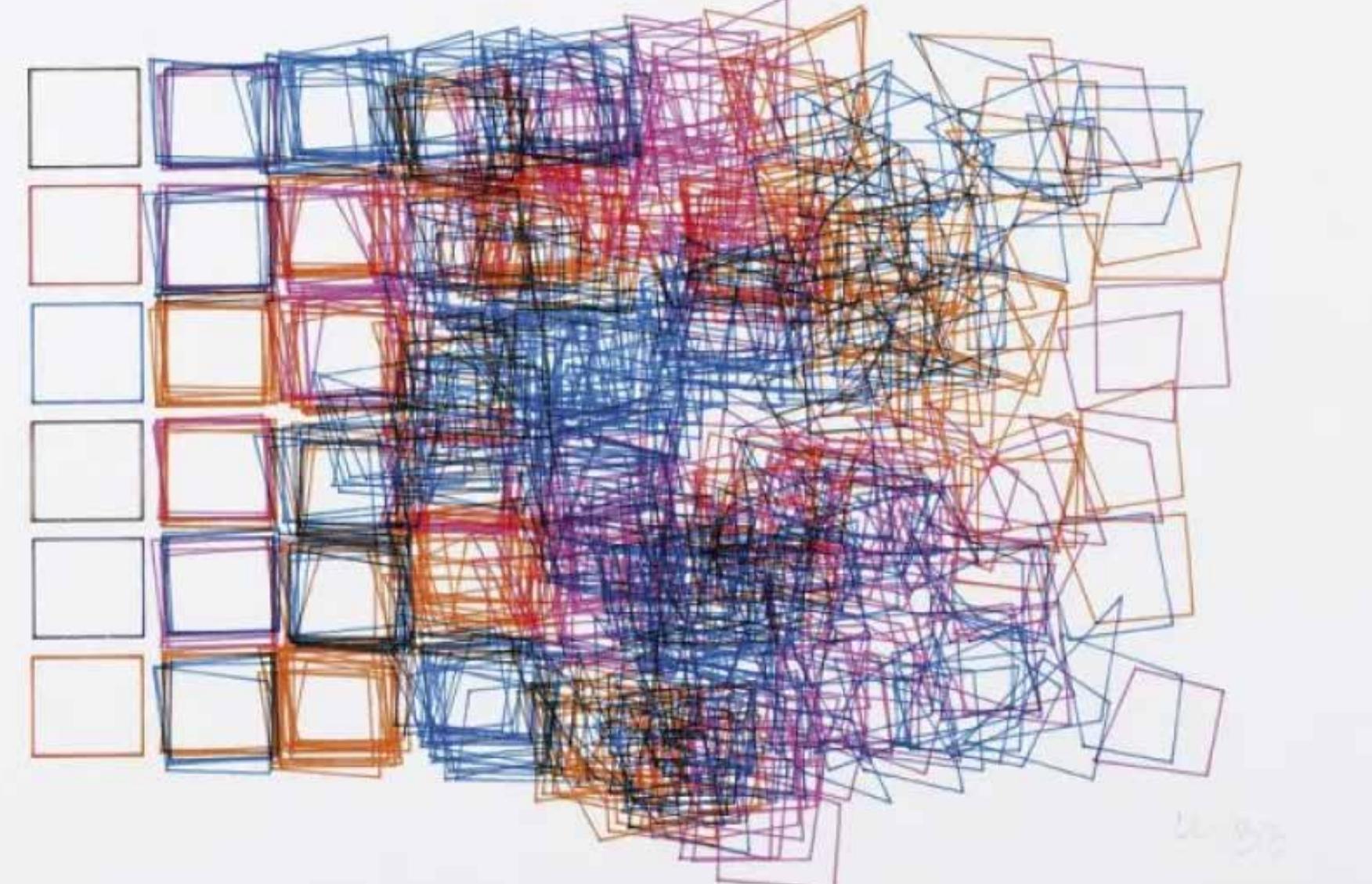
i will remove this //let distort = map(c, 0, columns - 1, 0, maxDistort);
 //takes the c number and in the range of c=0 to c=columns-1=10 it sets the distortion to the range of 0 to maxDistort=20

```
let columns = 11;
let rows = 7;
let space = 45;
let edgeX = 40;
let edgeY = 36;
let maxDistort = 20;
let repetitions = 10;

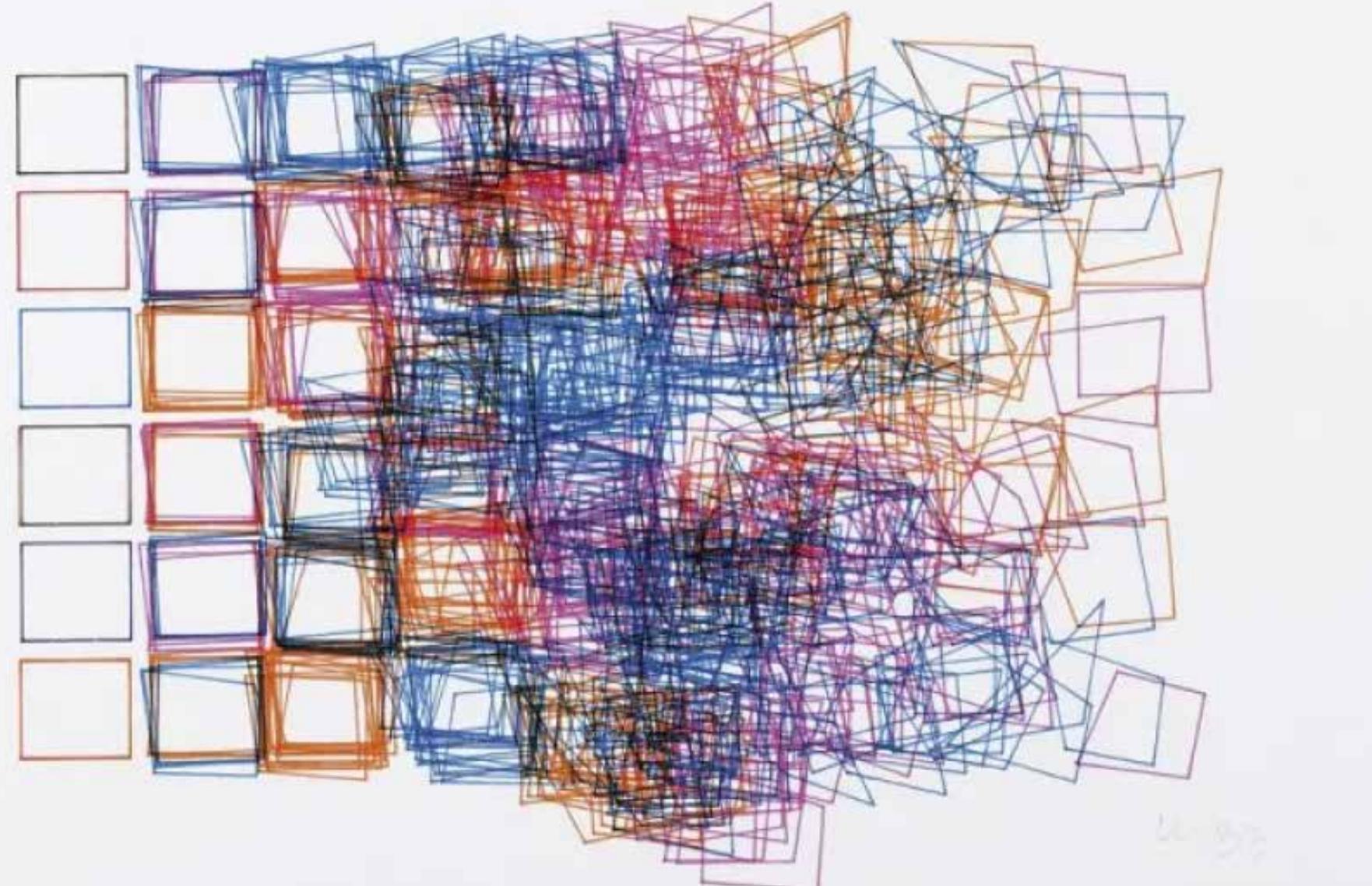
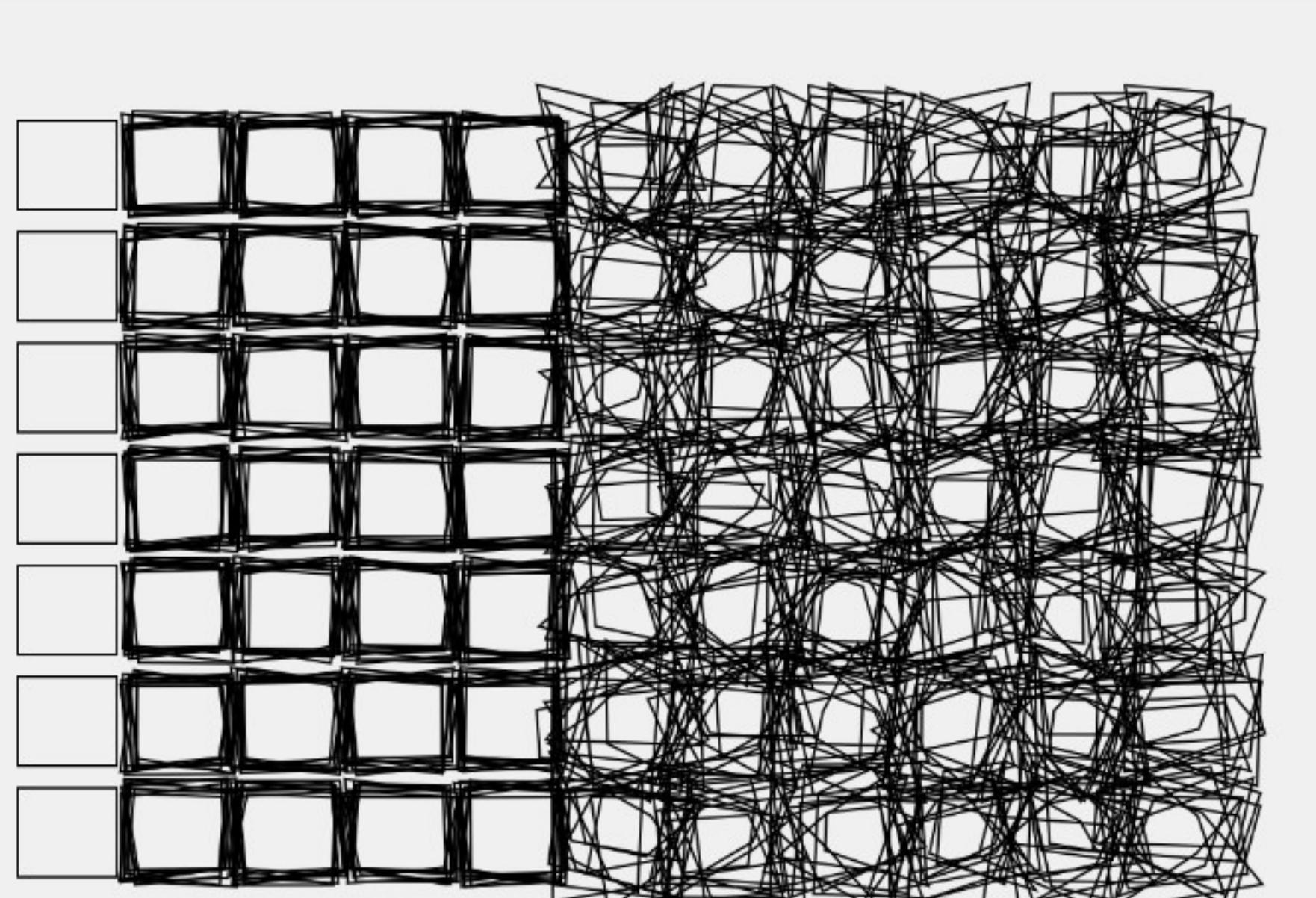
function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * space + 50;
      let startY = r * space + 50;
      let x1,x2,x3,x4,y1,y2,y3,y4
      if (c < 1) {
        x1 = startX;
        y1 = startY;
        x2 = startX + edgeX;
        y2 = startY;
        x3 = startX + edgeX;
        y3 = startY + edgeY;
        x4 = startX;
        y4 = startY + edgeY;
      }
      if (c >= 1 && c < 5){
        x1 = startX+random (-4,4)
        y1 = startY+random (-4,4);
        x2 = startX + edgeX+random (-4,4);
        y2 = startY+random (-4,4);
        x3 = startX + edgeX+random (-4,4);
        y3 = startY + edgeY+random (-4,4);
        x4 = startX+random (-4,4);
        y4 = startY + edgeY+random (-4,4);
      }
      for (let i = 0; i < repetitions; i++) {
        quad(x1, y1, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```

this resulted in the image "Step3_1". so there is randomness but all rectangles have the same coordinate in a stack
 (this makes sense as i didn't mention that they have to have different ones anywhere in my code)
 i will anyways also add the remaining columns to the code



```
added this  
if (c >= 5 && c < columns){  
    x1 = startX+random (-15,15)  
    y1 = startY+random (-15,15);  
    x2 = startX + edgeX+random (-15,15);  
    y2 = startY+random (-15,15);  
    x3 = startX + edgeX+random (-15,15);  
    y3 = startY + edgeY+random (-15,15);  
    x4 = startX+random (-15,15);  
    y4 = startY + edgeY+random (-15,15);  
}  
and got image "step3_2"  
so far so good
```



```
let columns = 11;
let rows = 7;
let space = 45;
let edgeX = 40;
let edgeY = 36;
let maxDistort = 20;
let repetitions = 10;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * space + 50;
      let startY = r * space + 50;
      let x1,x2,x3,x4,y1,y2,y3,y4

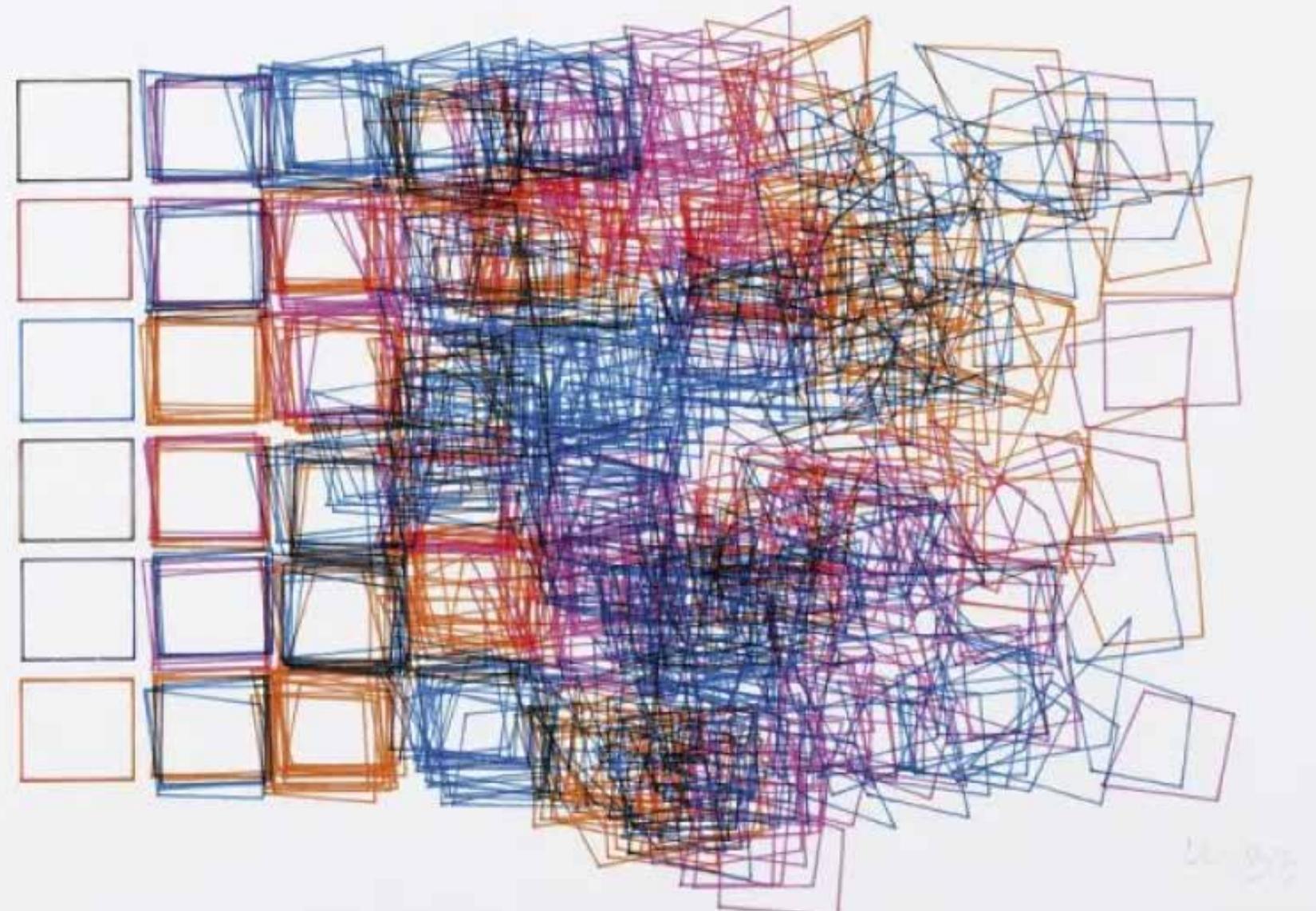
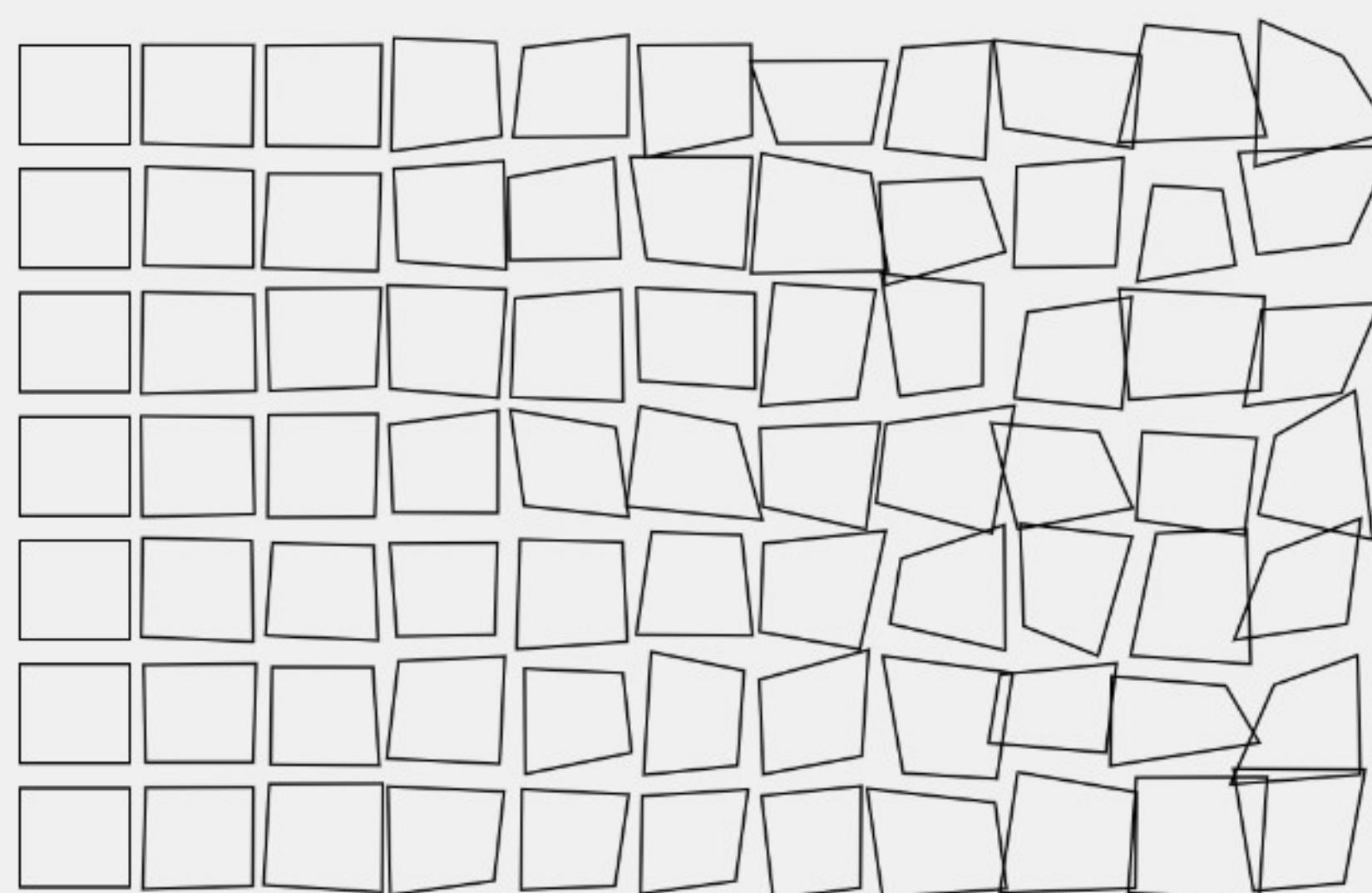
      for (let i = 0; i < repetitions; i++) {
        if (c < 1) {
          x1 = startX;
          y1 = startY;
          x2 = startX + edgeX;
          y2 = startY;
          x3 = startX + edgeX;
          y3 = startY + edgeY;
          x4 = startX;
          y4 = startY + edgeY;
        }

        if (c >= 1 && c < 5){
          x1 = startX+random (-4,4)
          y1 = startY+random (-4,4);
          x2 = startX + edgeX+random (-4,4);
          y2 = startY+random (-4,4);
          x3 = startX + edgeX+random (-4,4);
          y3 = startY + edgeY+random (-4,4);
          x4 = startX+random (-4,4);
          y4 = startY + edgeY+random (-4,4);
        }

        if (c >= 5 && c < columns){
          x1 = startX+random (-15,15)
          y1 = startY+random (-15,15);
          x2 = startX + edgeX+random (-15,15);
          y2 = startY+random (-15,15);
          x3 = startX + edgeX+random (-15,15);
          y3 = startY + edgeY+random (-15,15);
          x4 = startX+random (-15,15);
          y4 = startY + edgeY+random (-15,15);
        }

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

this was step 3_3. it resulted in each of the stack being distorted differently but the transition is not smooth because i used if statements for separate ranges



Step 4 Trying to make a smooth transition

```
let columns = 11;
let rows = 7;
let space = 45;
let edgeX = 40;
let edgeY = 36;
let maxDistort = 20;
let repetitions = 10;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

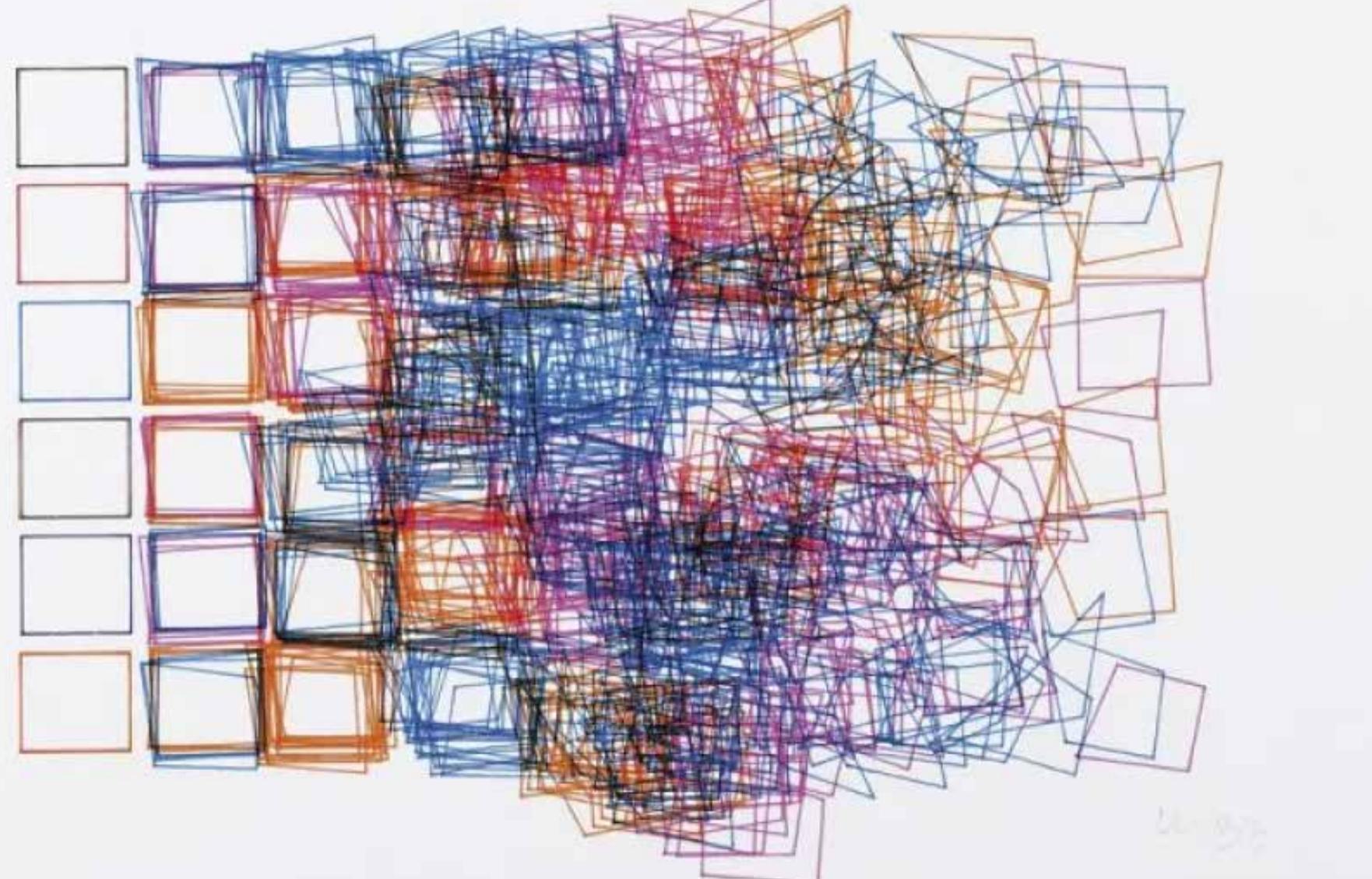
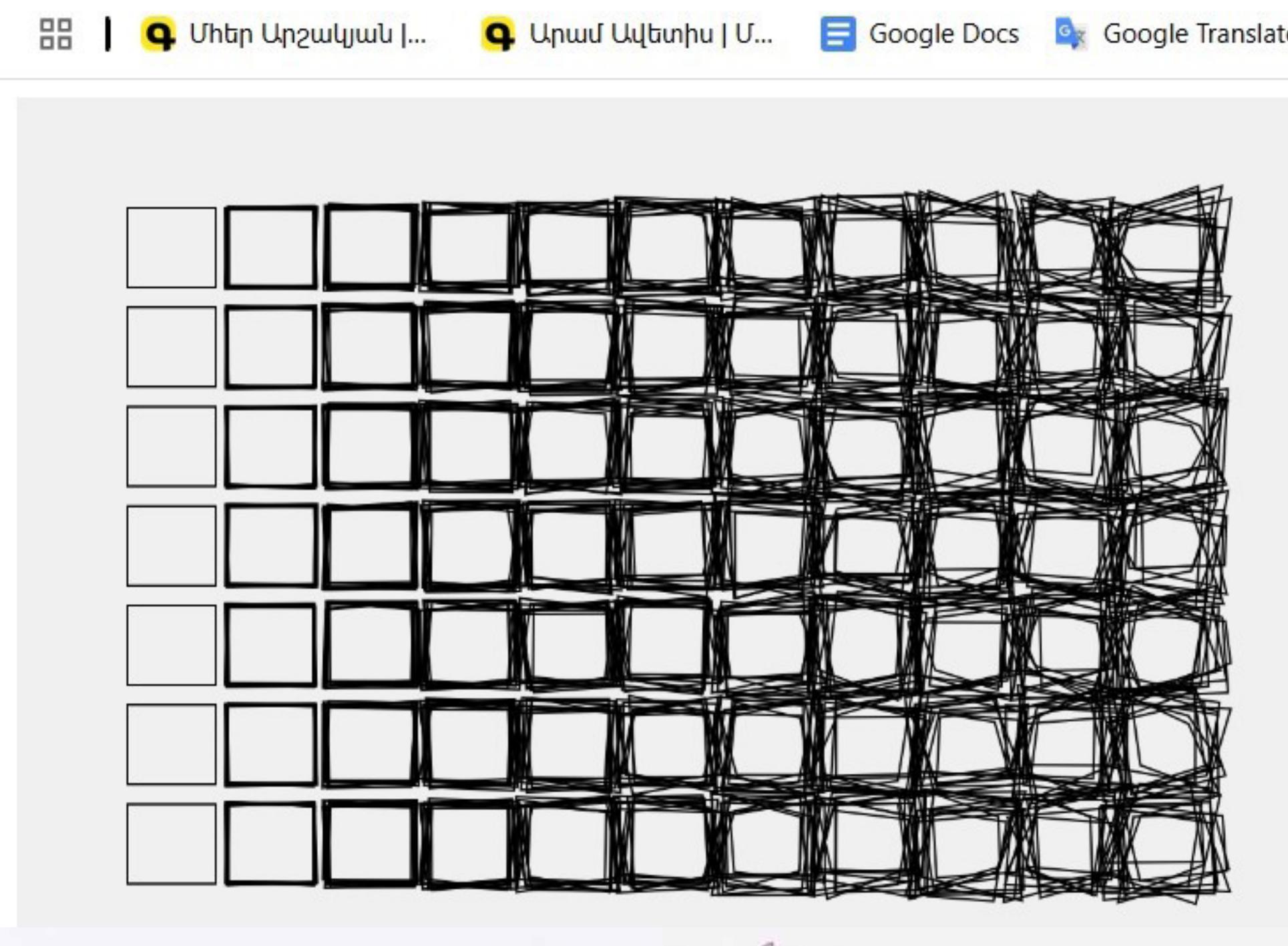
  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * space + 50;
      let startY = r * space + 50;
      let x1,x2,x3,x4,y1,y2,y3,y4

      for (let i = 0; i < repetitions; i++) {

        x1 = startX+random (-c,c)
        y1 = startY+random (-c,c);
        x2 = startX + edgeX+random (-c,c);
        y2 = startY+random (-c,c);
        x3 = startX + edgeX+random (-c,c);
        y3 = startY + edgeY+random (-c,c);
        x4 = startX+random (-c,c);
        y4 = startY + edgeY+random (-c,c);
      }

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

the transition is smoother (compare with step3_2) but the stacked rectangles are the same again
```

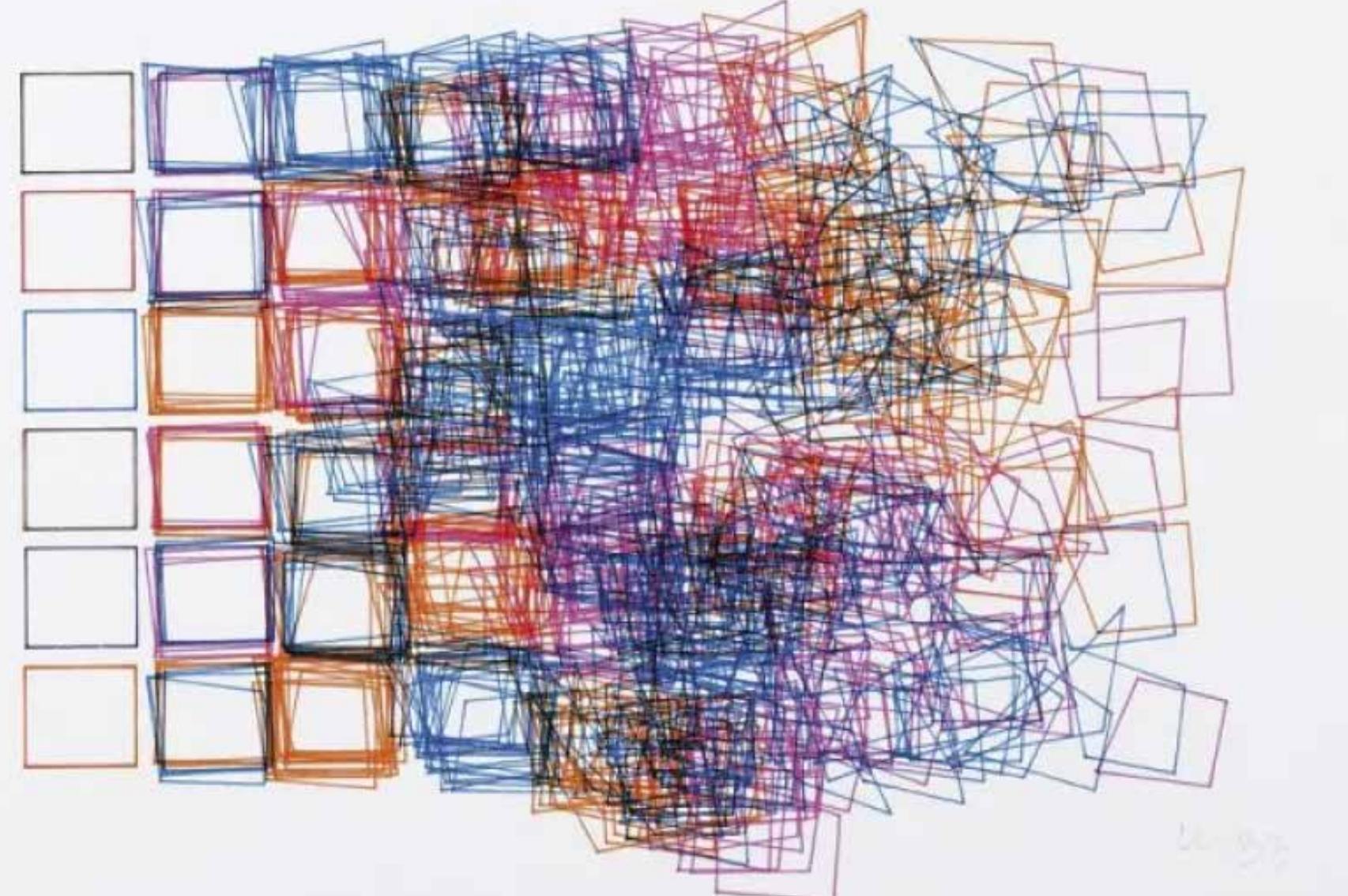
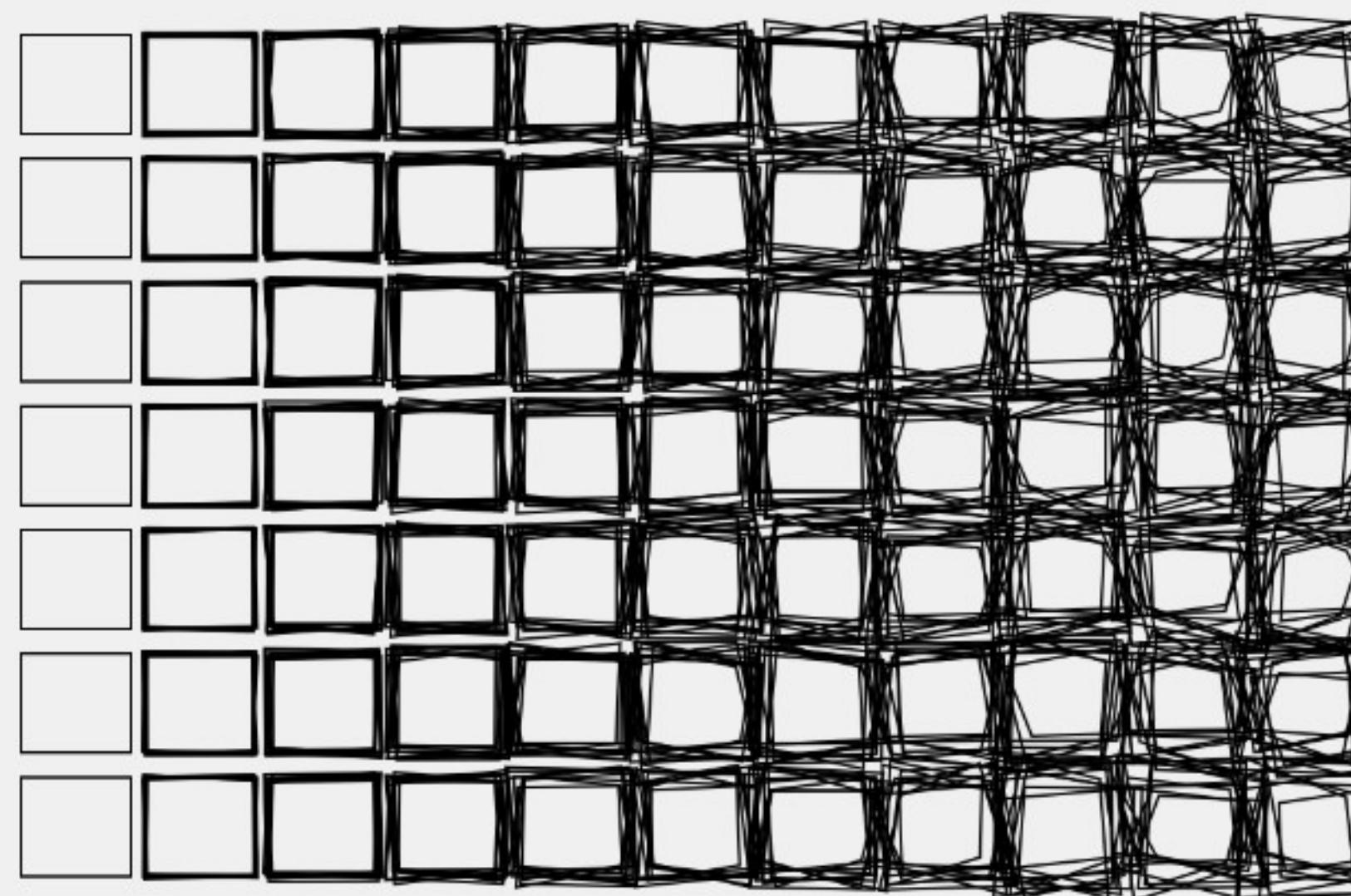


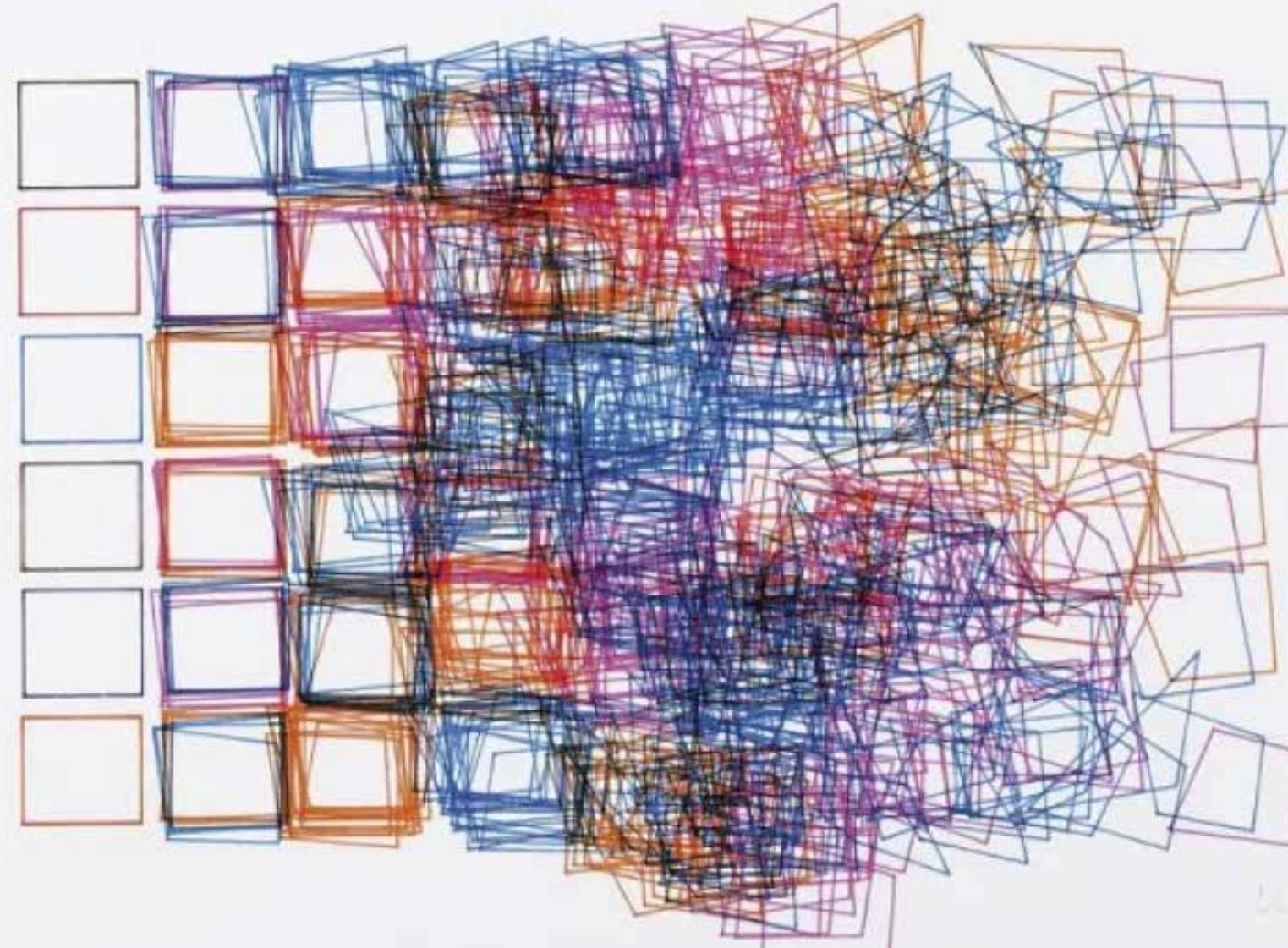
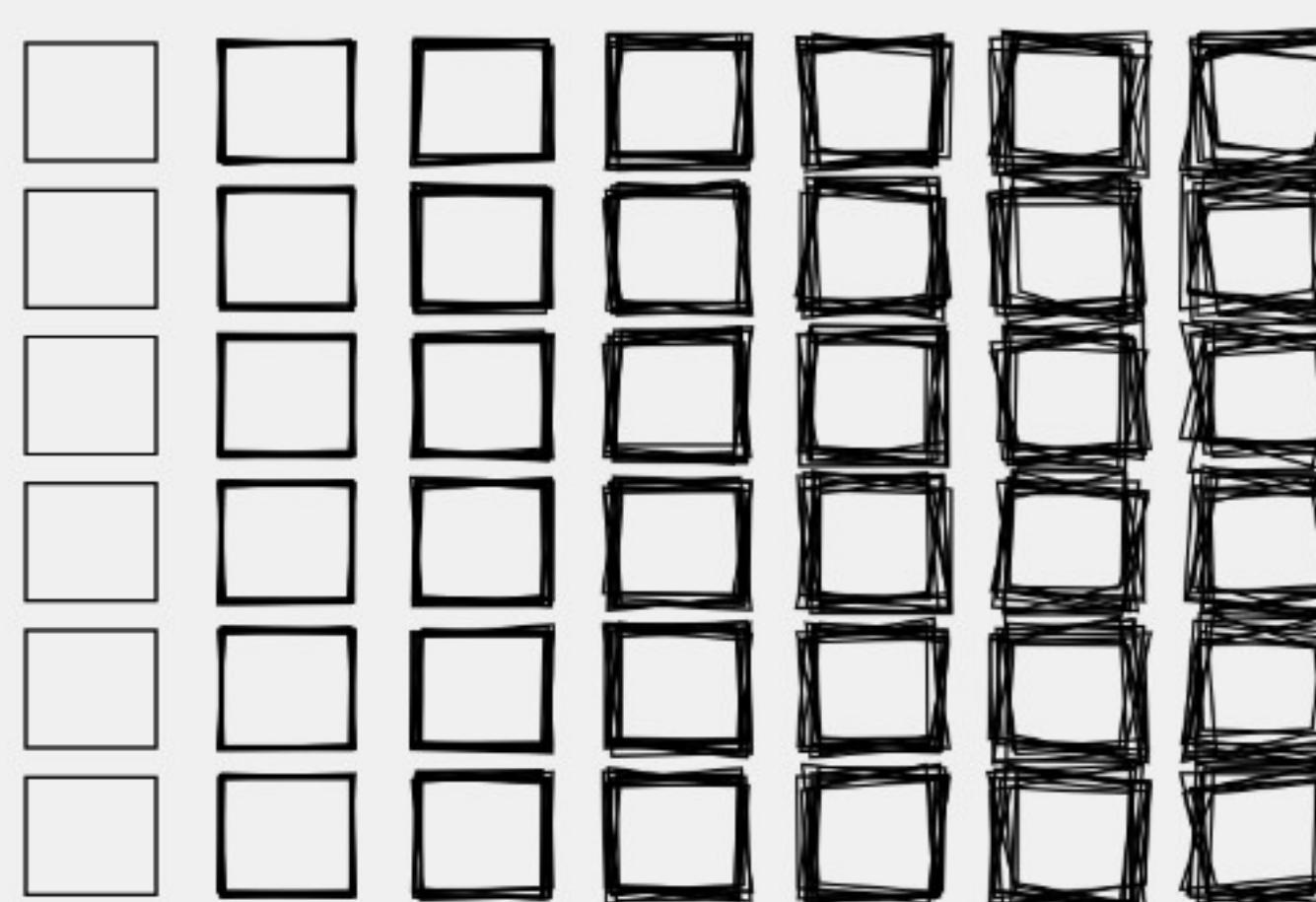
haha, i again forgot to place the quad inside the loop before.

now the smoothness is actually perfect, i will just make the distortion more intense by changing the randomness range from $(-c, c)$ to $(-c+5, c+5)$

5 was too small, i increased the number to 8 and i got step4_2

```
x1 = startX+random (-c+8,c+8);
y1 = startY+random (-c+8,c+8);
x2 = startX + edgeX+random (-c+8,c+8);
y2 = startY+random (-c+8,c+8);
x3 = startX + edgeX+random (-c+8,c+8);
y3 = startY + edgeY+random (-c+8,c+8);
x4 = startX+random (-c+8,c+8);
y4 = startY + edgeY+random (-c+8,c+8);
```





Step 5 reduction of the number of rows and intensifying the distortion

```
let columns = 7;
let rows = 6;
let spaceX = 60;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let maxDistort = 20;
let repetitions = 10;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

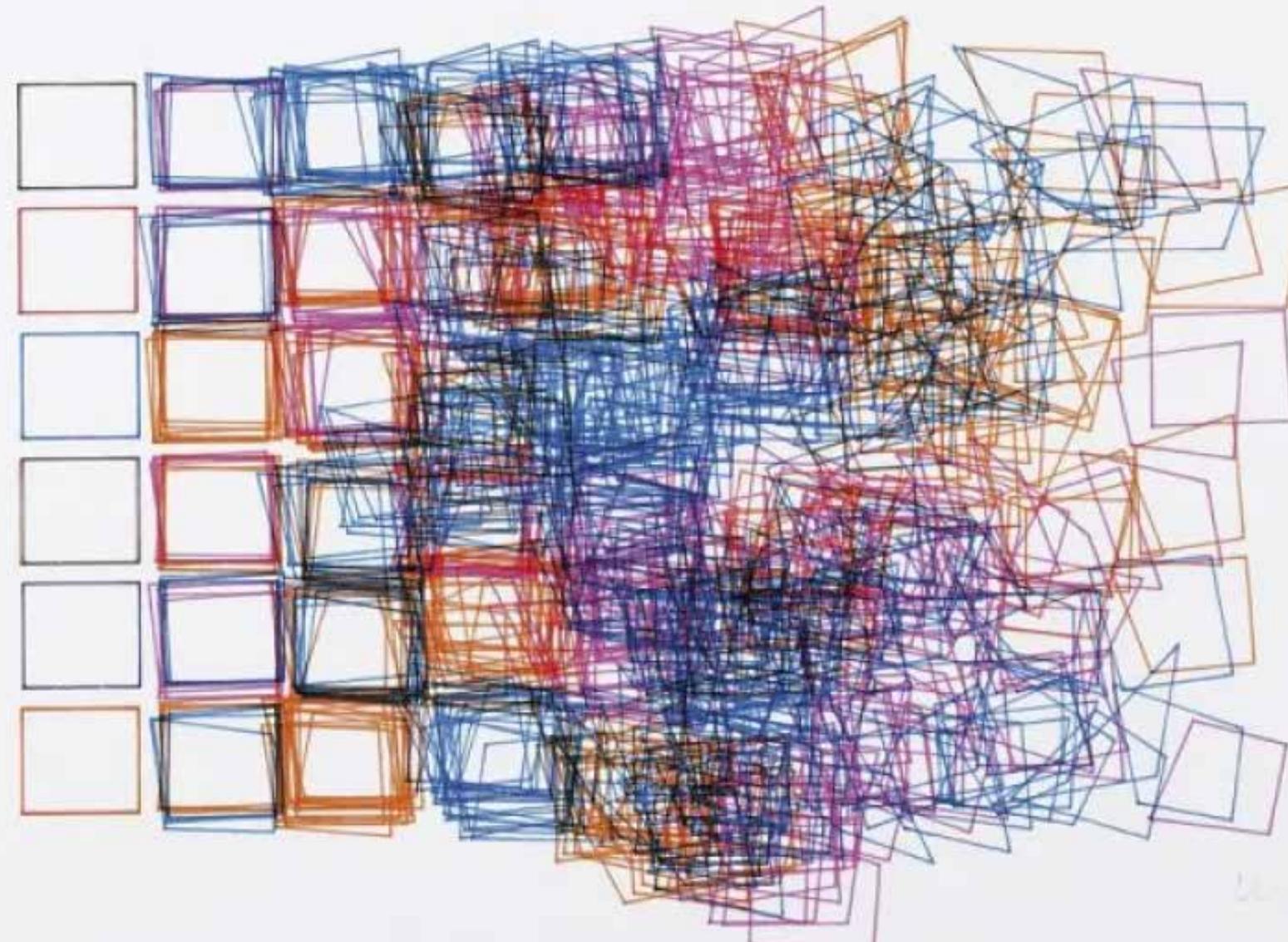
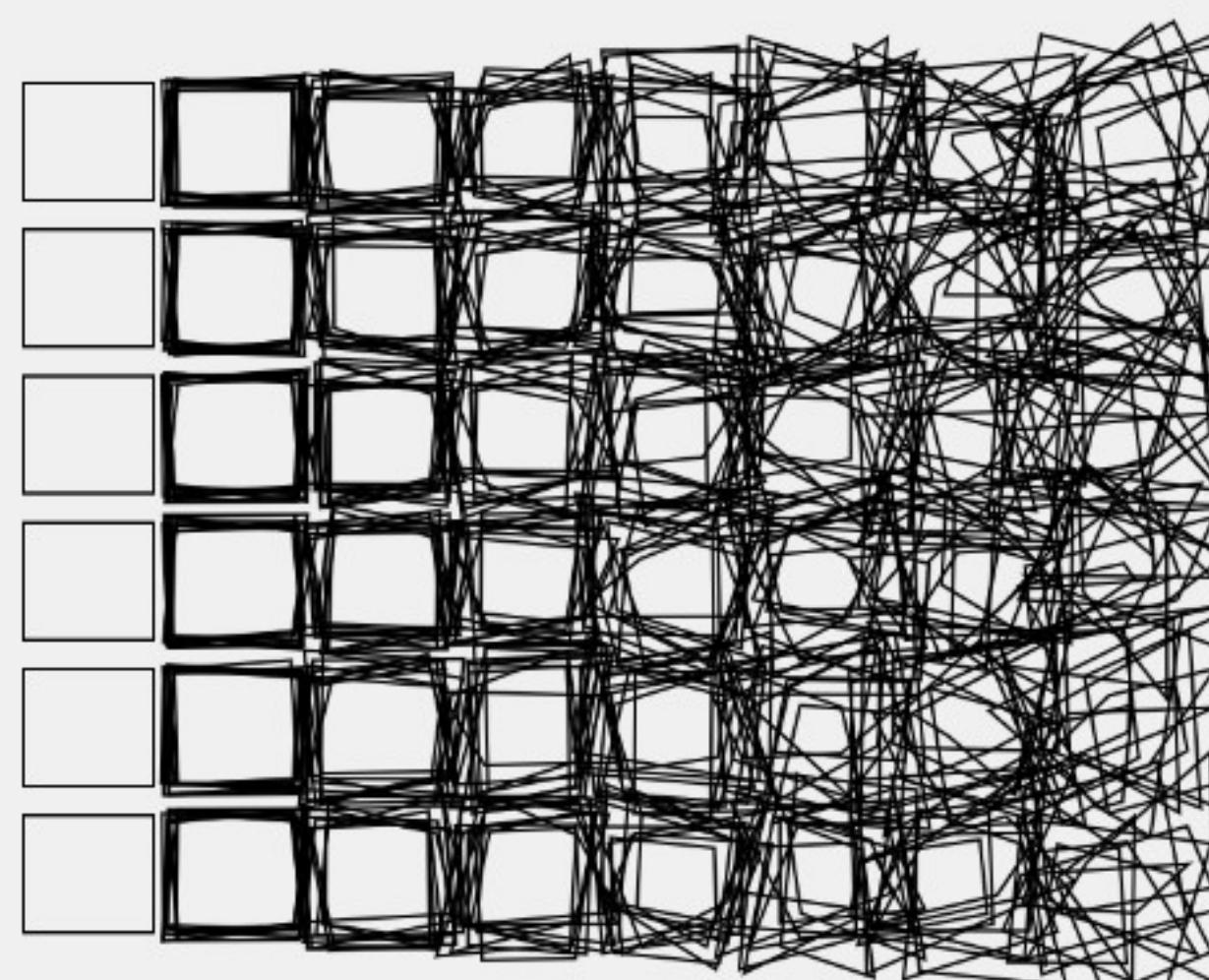
  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * spaceX ;
      let startY = r * spaceY ;
      let x1,x2,x3,x4,y1,y2,y3,y4

      for (let i = 0; i < repetitions; i++) {

        x1 = startX+random (-c+100,c+100);
        y1 = startY+random (-c+100,c+100);
        x2 = startX + edgeX+random (-c+100,c+100);
        y2 = startY+random (-c+100,c+100);
        x3 = startX + edgeX+random (-c+100,c+100);
        y3 = startY + edgeY+random (-c+100,c+100);
        x4 = startX+random (-c+100,c+100);
        y4 = startY + edgeY+random (-c+100,c+100);

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

this was step 5_1, i am not satisfied with the intensity
```



i switched to mapping so that the distortion is in a linear relation to the column number

```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 20;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = c * spaceX + 50;
      let startY = r * spaceY + 50;

      let distort = map(c, 0, columns - 1, 0, maxDistort);

      for (let i = 0; i < repetitions; i++) {
        let xl = startX + random(-distort, distort);
        let yl = startY + random(-distort, distort);

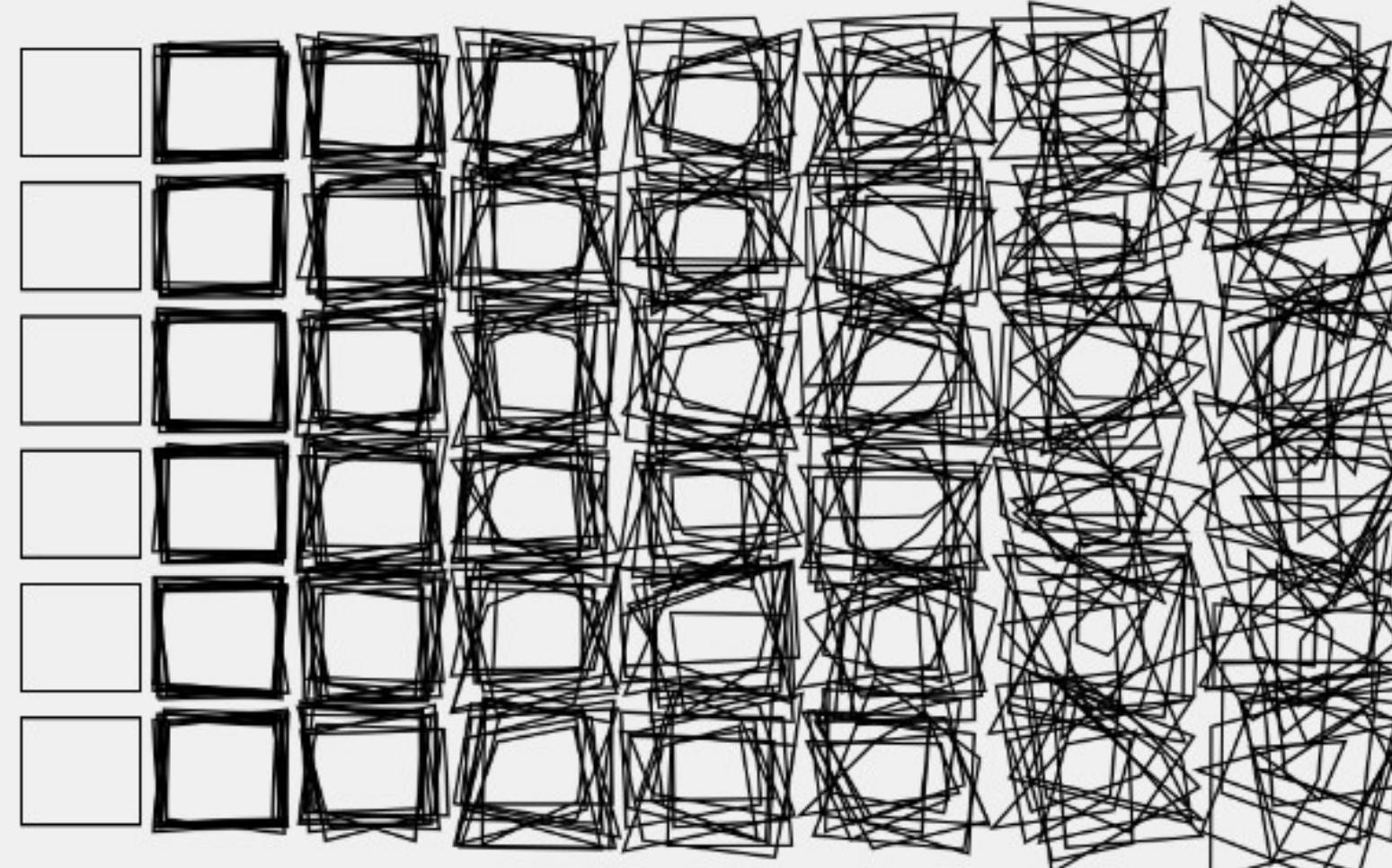
        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + edgeY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

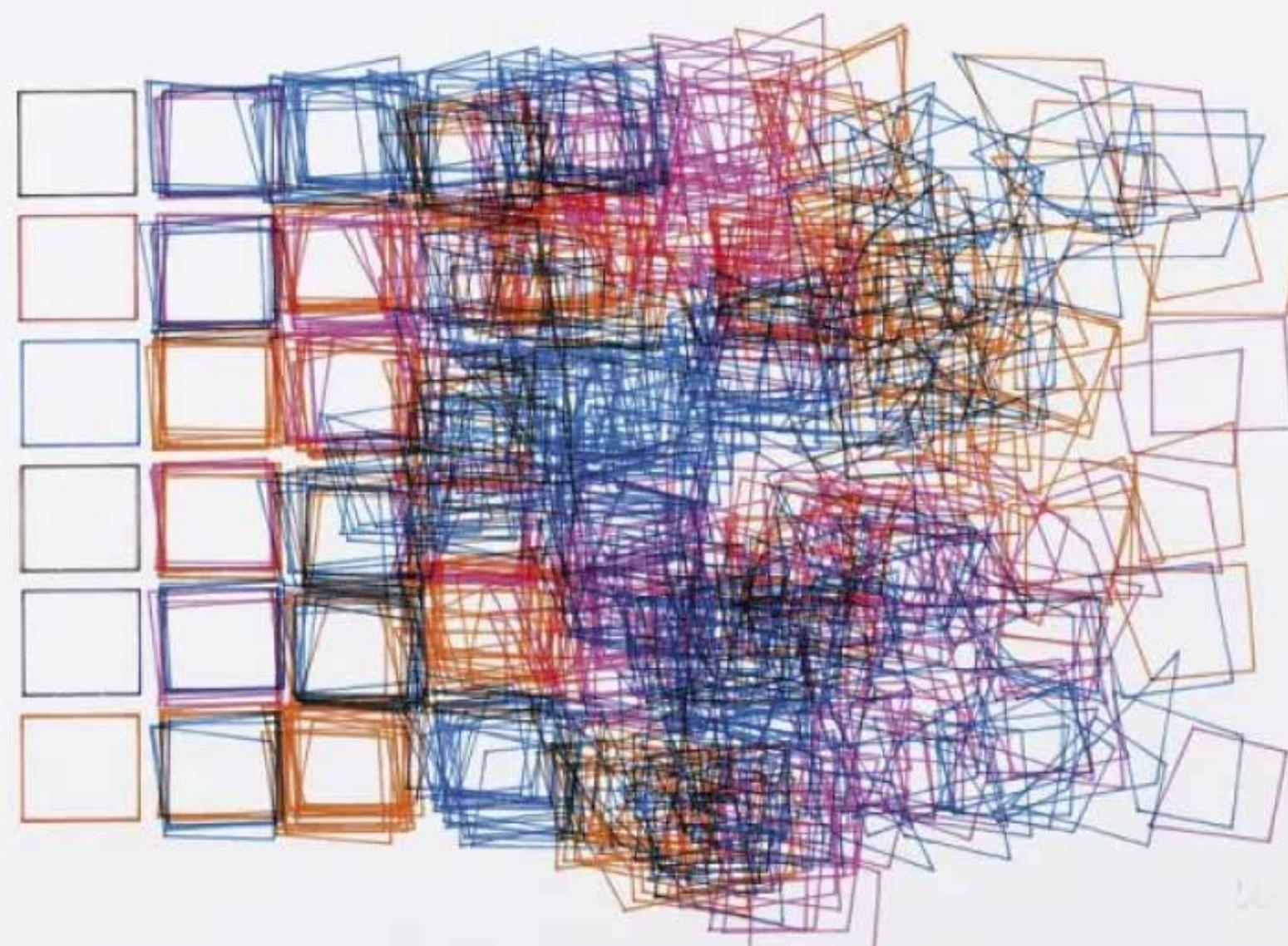
        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

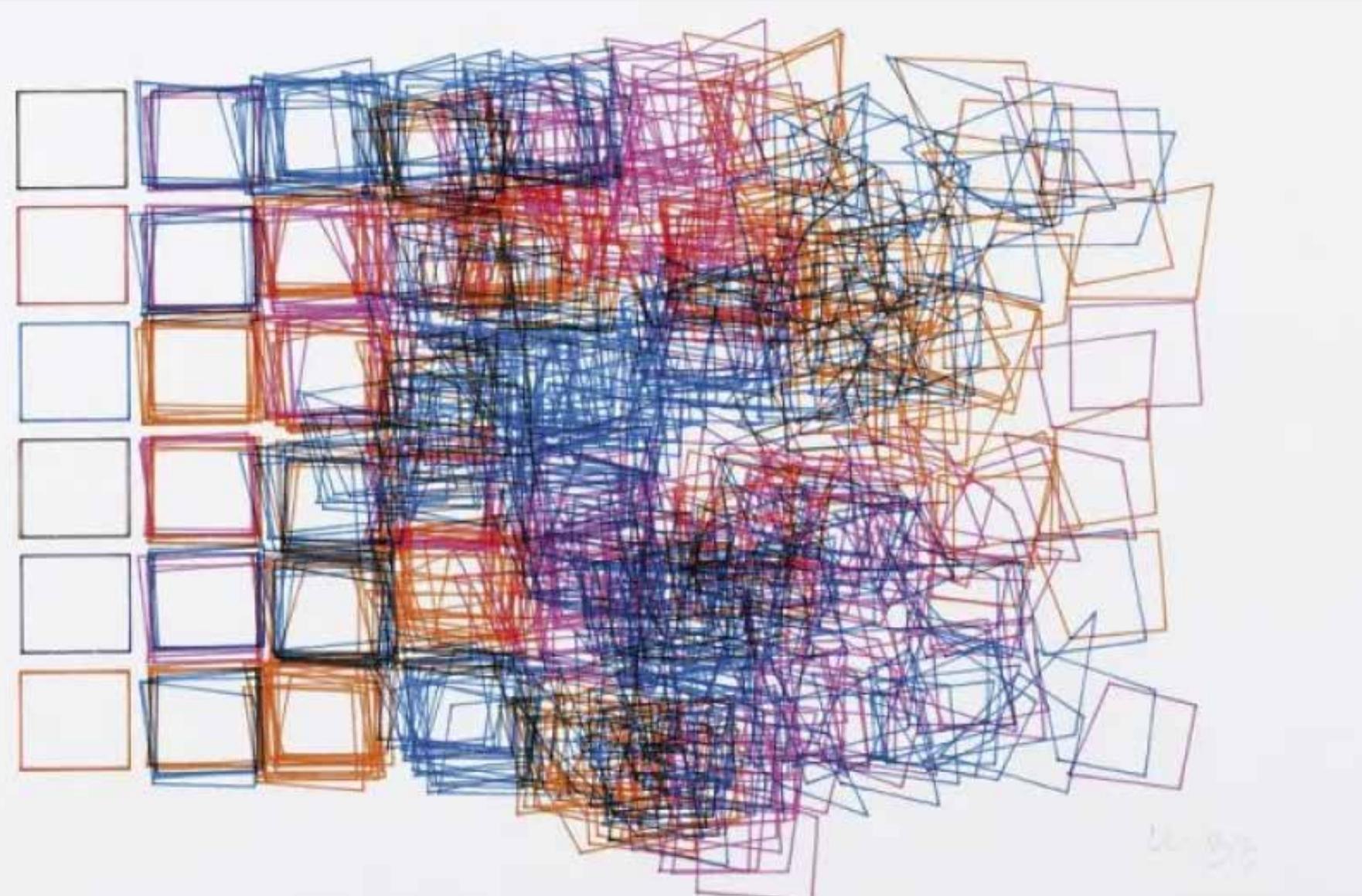
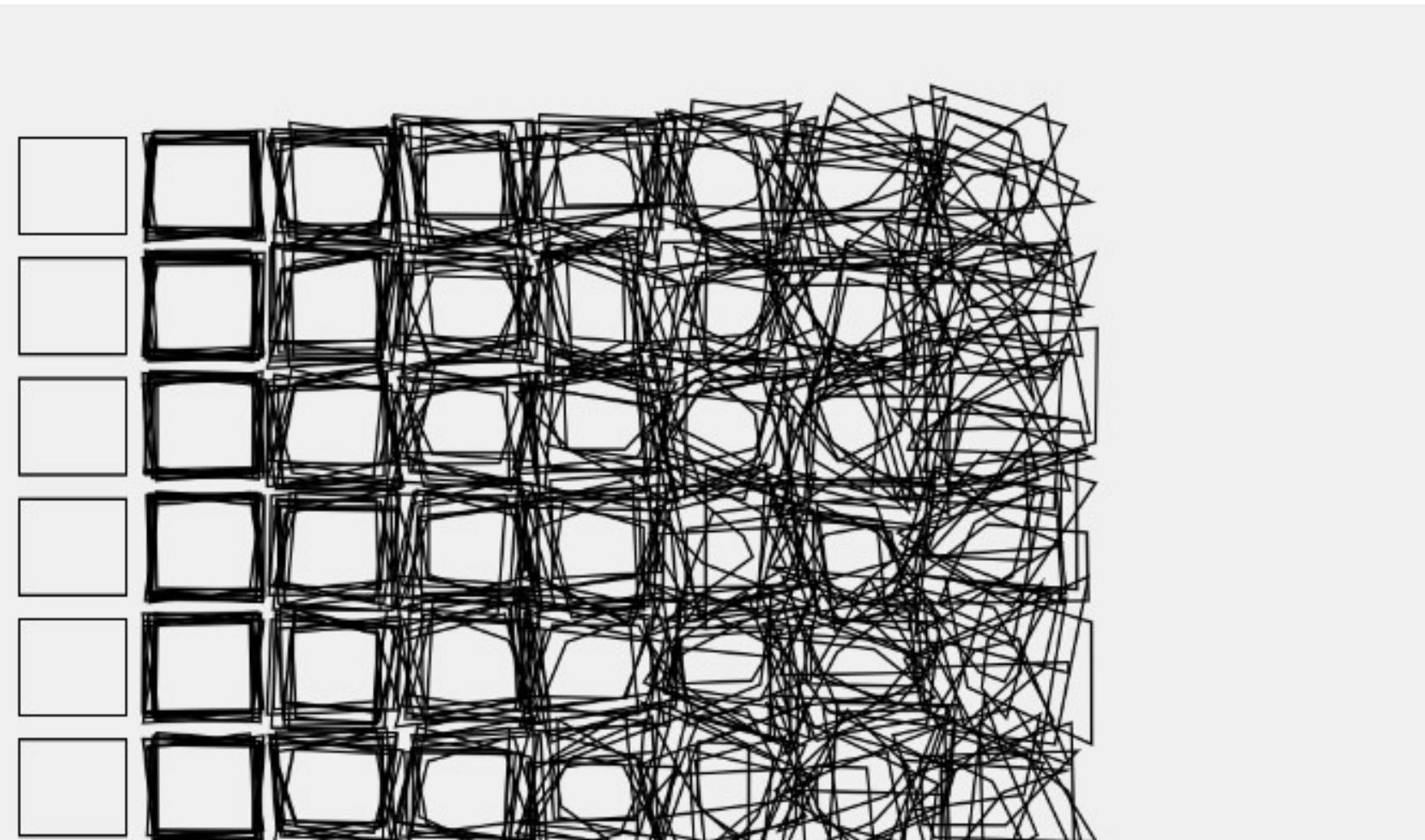
        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```

i think my width is off. it seems like the spacing should be adjusted but other than that, this (step 5_2 doesn't look very bad)



spacing variations
non-linear function
let startX = 50 + c * (spaceX + c * 2) (step 5_3)





```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 20;

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  stroke(0);
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distort = map(c, 0, columns - 1, 0, maxDistort);

      for (let i = 0; i < repetitions; i++) {
        let xl = startX + random(-distort, distort);
        let yl = startY + random(-distort, distort);

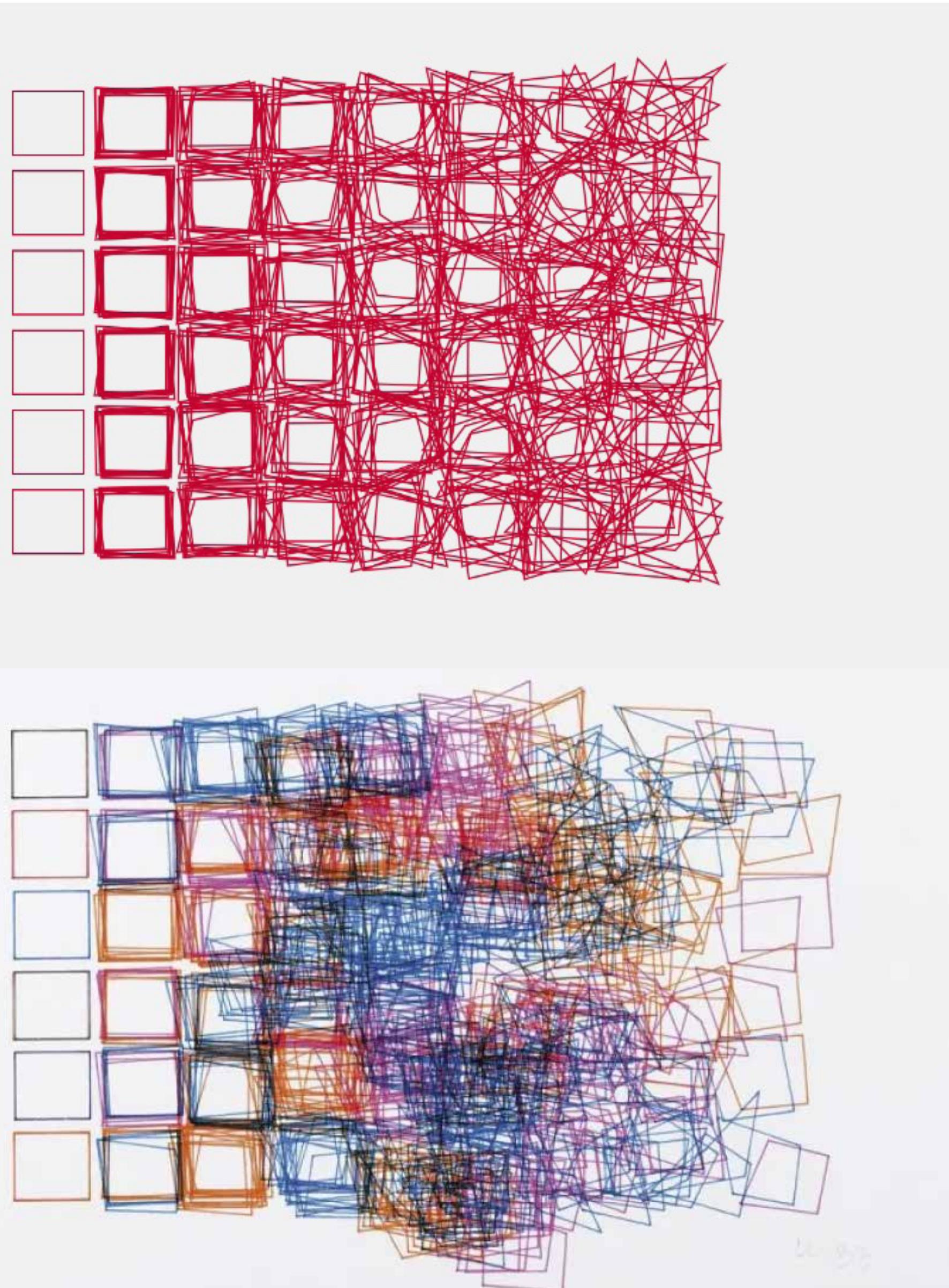
        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}

^ this was step 5_4
closer but not quite there. i think with molnar's artwork, the distortion is stronger in
the middle and weaker towards the edges.
but i was thinking of getting to that part with animation and not a still
```



Step 6 random color combinations

```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 20;
let palette = ['#16161B', '#D37139', '#C2042E', '#3C60AA', '#9A3E7D'];

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  let chosenColor = random(palette);
  stroke(chosenColor);
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distort = map(c, 0, columns - 1, 0, maxDistort);

      for (let i = 0; i < repetitions; i++) {
        let xl = startX + random(-distort, distort);
        let yl = startY + random(-distort, distort);

        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + edgeY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

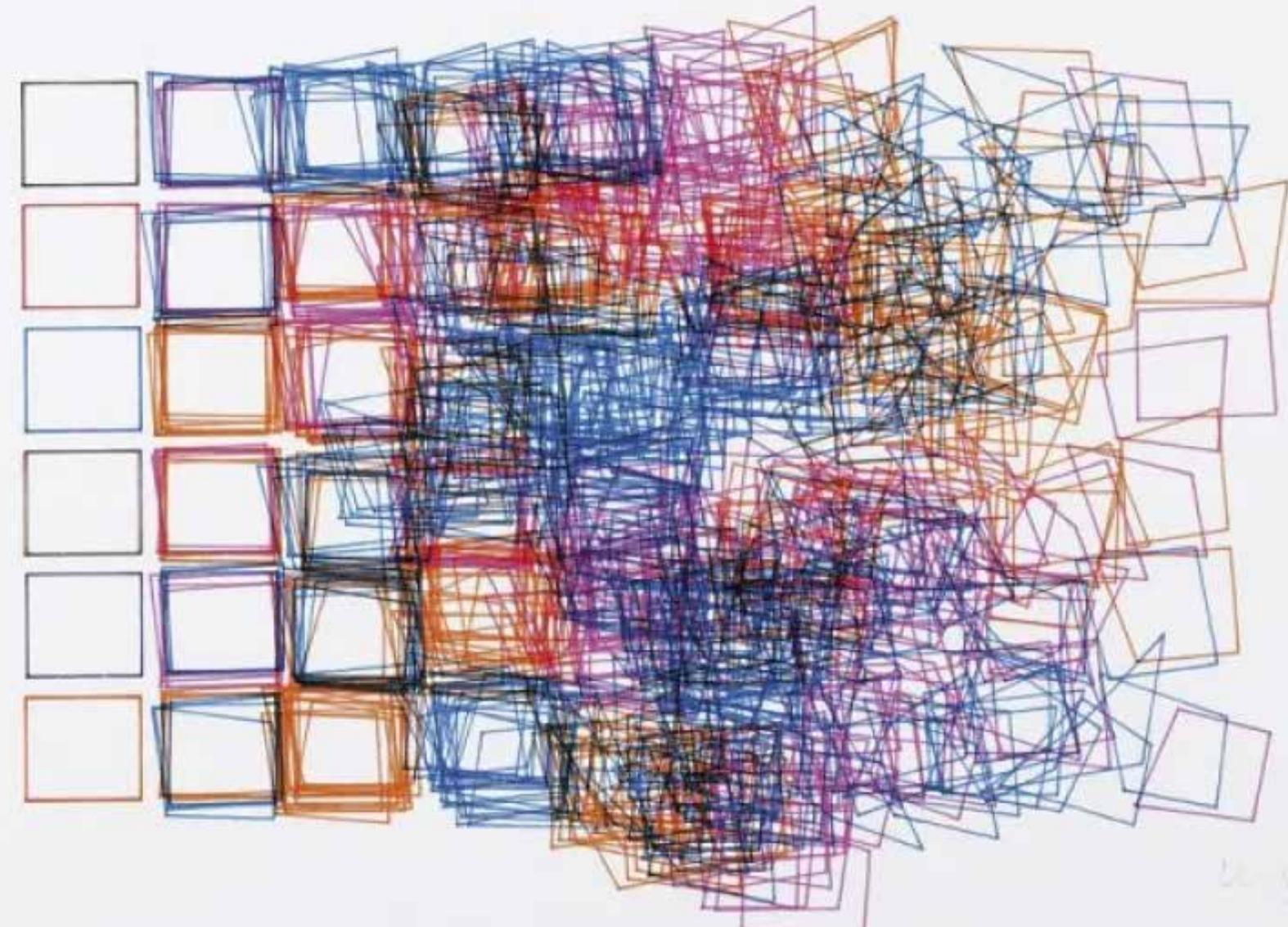
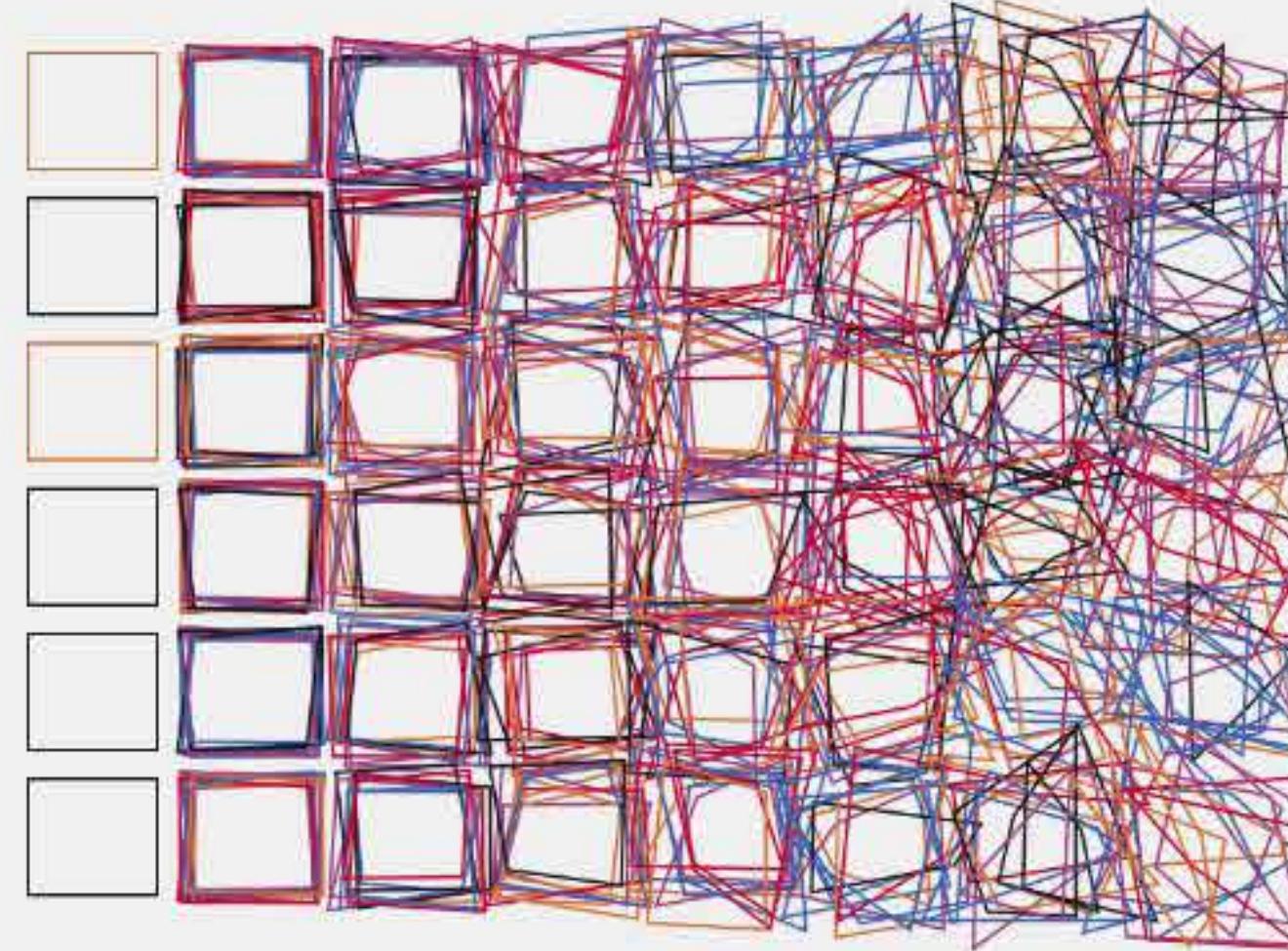
        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```

it is now all just red hmmmm. i guess i forgot to assign it to each quad being created, maybe the placement is wrong again

127.0.0.1:5500

All Bookmarks



```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 20;
let palette = ['#16161B', '#D37139', '#C2042E', '#3C60AA', '#9A3E7D'];

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distort = map(c, 0, columns - 1, 0, maxDistort);

      for (let i = 0; i < repetitions; i++) {
        let xl = startX + random(-distort, distort);
        let yl = startY + random(-distort, distort);

        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        let chosenColor = random(palette);
        stroke(chosenColor);
        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}

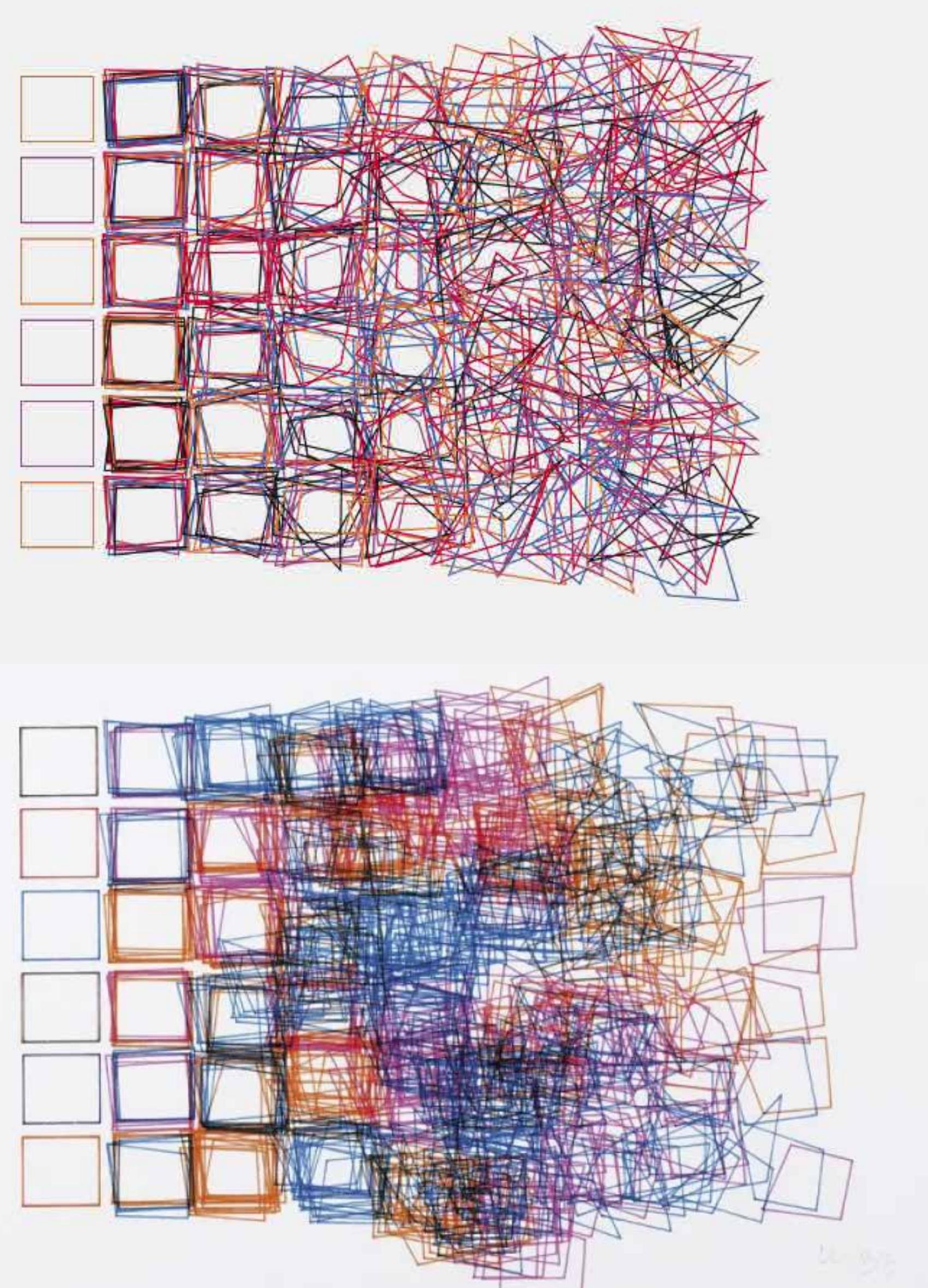
i was right and i managed to correct this mistake without external help!
```

[Ահեղ Արշակյան |...](#)[Արամ Ավետիսի | U...](#)[Google Docs](#)[Google Translate](#)[Watch Four Rooms \(...\)](#)[A Supposedly Fun T...](#)[Looking for a place...](#)[Բառացանկ – այլե...](#)

»

All Bookmarks

Step 7 more distortion tryouts



```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 30;
let palette = ['#16161B', '#D37139', '#C2042E', '#3C60AA', '#9A3E7D'];

function setup() {
  createCanvas(600, 400);
  background(240);
  noFill();
  strokeWeight(1);

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distort = map(c, 0, columns - 1, 0, maxDistort);

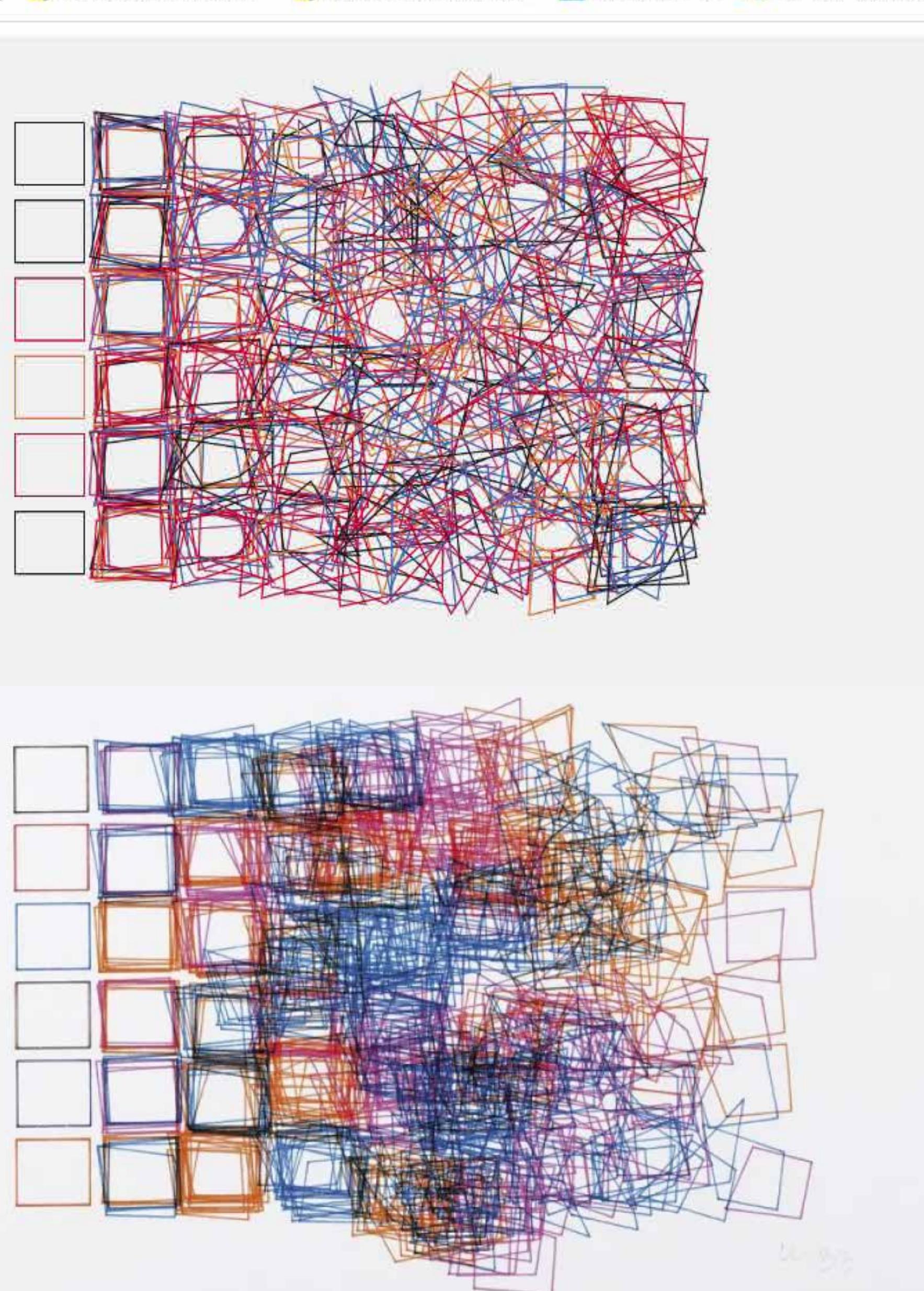
      for (let i = 0; i < repetitions; i++) {
        let x1 = startX + random(-distort, distort);
        let y1 = startY + random(-distort, distort);

        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        let chosenColor = random(palette);
        stroke(chosenColor);
        quad(x1, y1, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```



Step 8 Animating

```
let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 20;
let palette = ['#16161B', '#D37139', '#C2042E',
  '#3C60AA', '#9A3E7D'];

function setup() {
  createCanvas(600, 400);
  noFill();
  strokeWeight(1);
}

function draw() {
  background(240); // clear previous frame

  let hoveredColumn = floor((mouseX - 50) /
    (spaceX + 4)); // get hovered column

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distort;
      if (c === hoveredColumn) {
        distort = maxDistort;
      } else {
        distort = 0;
      }

      for (let i = 0; i < repetitions; i++) {
        let x1 = startX + random(-distort, distort);
        let y1 = startY + random(-distort, distort);

        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + edgeY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        let chosenColor = random(palette);
        stroke(chosenColor);
        quad(x1, y1, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}
```

```

i will anyways add the neighboring columns as well

let columns = 8;
let rows = 6;
let spaceX = 45;
let spaceY = 45;
let edgeX = 40;
let edgeY = 36;
let repetitions = 10;
let maxDistort = 30;
let palette = ['#16161B', '#D37139', '#C2042E', '#3C60AA', '#9A3E7D'];

function setup() {
  createCanvas(600, 400);
  noFill();
  strokeWeight(1);
}

function draw() {
  background(240);

  let hoveredColumn = floor((mouseX - 50) / (spaceX + 4));

  for (let r = 0; r < rows; r++) {
    for (let c = 0; c < columns; c++) {
      let startX = 50 + c * (spaceX + 4);
      let startY = r * spaceY + 50;

      let distance = abs(c - hoveredColumn);
      let distort = 0;

      if (distance <= 5) {
        distort = map(distance, 0, 5, maxDistort, 0);
      }

      for (let i = 0; i < repetitions; i++) {
        let xl = startX + random(-distort, distort);
        let yl = startY + random(-distort, distort);

        let x2 = startX + edgeX + random(-distort, distort);
        let y2 = startY + random(-distort, distort);

        let x3 = startX + edgeX + random(-distort, distort);
        let y3 = startY + edgeY + random(-distort, distort);

        let x4 = startX + random(-distort, distort);
        let y4 = startY + edgeY + random(-distort, distort);

        let chosenColor = random(palette);
        stroke(chosenColor);
        quad(xl, yl, x2, y2, x3, y3, x4, y4);
      }
    }
  }
}

```