

Code and Generativity

Generativity

- Rules
- Repetition, but with variation
- Emergence
 - Unpredictability
 - Even the simplest code when it emerges, the result becomes unimaginable
 - A logical outcome of the code that even the programmer did not foresee

random()

Finder et tilfældigt decimaltal*:

Mellem 0 og x:

```
var randX = random(x);
```

Mellem x(inklusiv) og y(eksklusiv):

```
var randXY = random(x,y);
```

Eller et random element i array:

```
var myArray = [...];
```

```
var randArray = random(myArray);
```

*brug int() eller floor() til at
nedrunde til nærmeste heltal

Reference: <https://p5js.org/reference/#/p5/random>

noise()

Finder en række tal der er mere "harmoniske" ift. hinanden end med random()

Returnerer altid et tal mellem 0.0 og 1.0

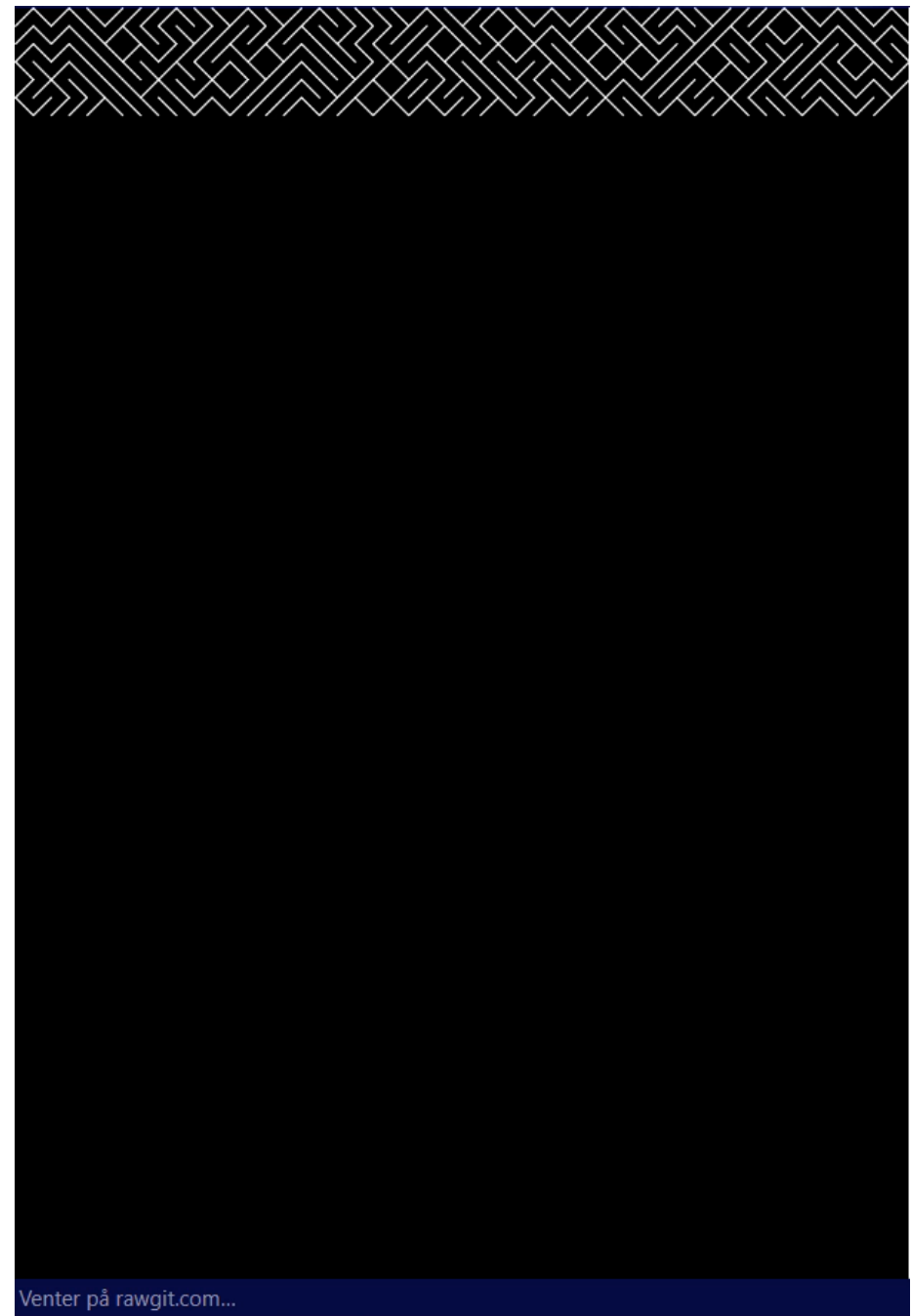
```
function draw() {  
  ...  
  y += 0.01;  
  var x = noise(y);  
  ...  
}
```

Jo mindre variation for hvert kald til noise, jo mere "harmoniske".

Reference: <https://p5js.org/reference/#/p5/noise>

Winnie's code

```
8 Rules:
9 1. Throw a dice randomly
10 2. Half of the chance print a backward slash
11 3. The other half of the chance will print a forward slash
12 4. Fill the whole canvas with the randomly generated slash
13 invent your own 10Print by creating your own rules!
14 */
15
16 let x = 0;
17 let y = 0;
18 let spacing = 10;
19
20 function setup() {
21   createCanvas(500,800);
22   background(0);
23 }
24
25 function draw() {
26   stroke(255);
27   if (random(1) < 0.5) { //probability
28     //line(0,0,10,10); //backward slash
29     line(x,y,x+spacing,y+spacing);
30   } else {
31     //line(0,10,10,0); //forward slash
32     line(x,y+spacing,x+spacing,y); //forward slash
33   }
34   x+=10;
35   if (x > width) {
36     x = 0;
37     y += spacing;
38   }
39 }
40 }
```



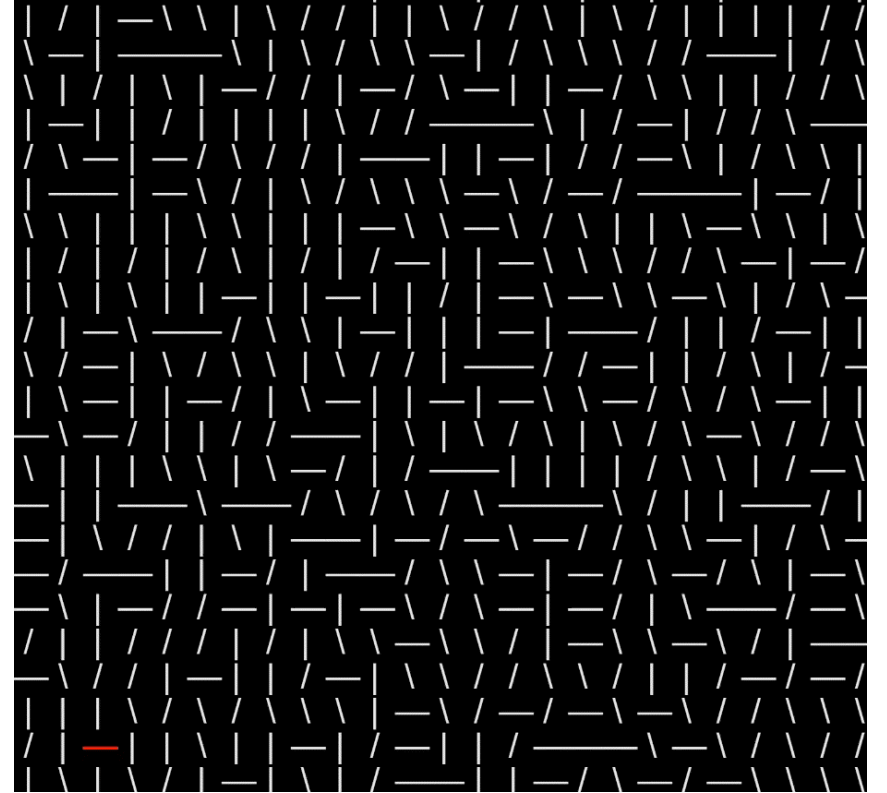
```

48 class Throbber{
49     constructor(no, xpos, ypos) {
50         this.no = no;
51         this.pos = new createVector(xpos, ypos);
52     }
53
54     display(state) {
55         if (state) {
56             fill (255,0,0);
57         } else {
58             fill(230);
59         }
60         textSize(30);
61         text(slash[this.no], this.pos.x, this.pos.y);
62     }
63
64     doThrob() {
65         this.no++;
66         if (this.no == slash.length) {
67             this.no = 0;
68         }
69     }
70 }

```

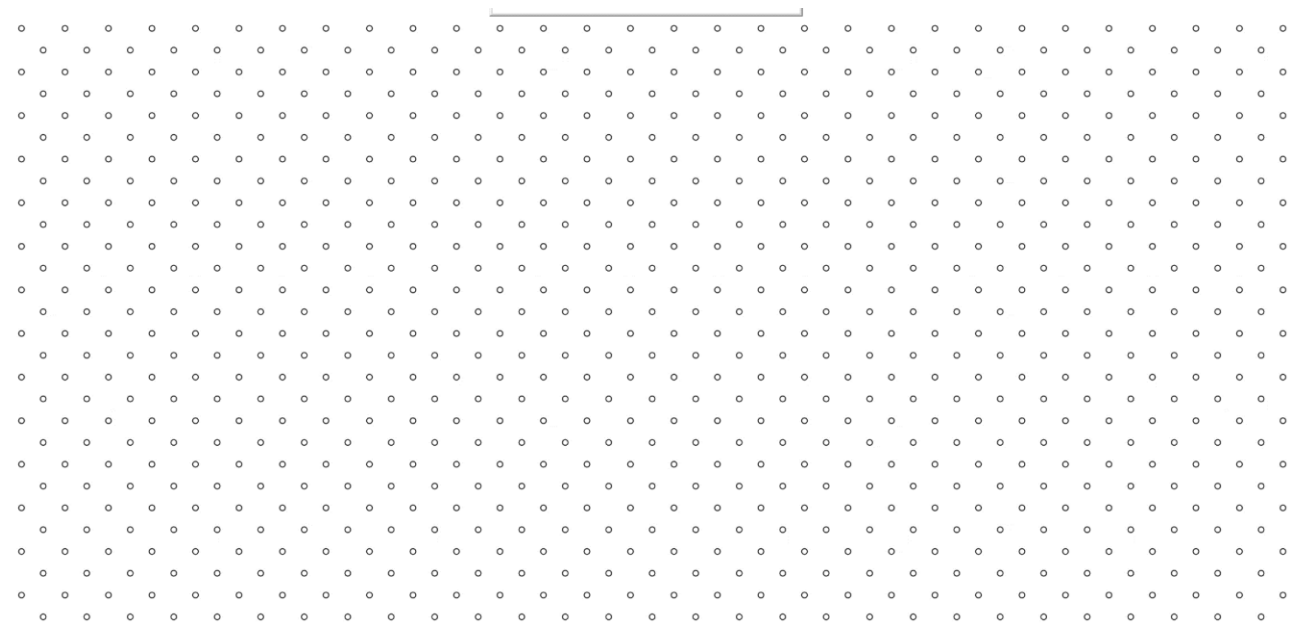
andom throbber to rotate
));

);



What are the rules?

```
22  if (status == "up") {
23      r+=0.01;    //increase the size at a time
24  }else{
25      r-=0.01;
26  }
27
28  //if it reaches certain size, change the direction
29  if (r > 40){    //until r reaches a certain number, reset the status
30      status = "down";
31  }else if (r < 0) {
32      status = "up";
33  }
34  console.log(r);
35  }
36
37  function drawSomething(x, y) {    // total horizontal ellipses (x axis =
38      ellipse(x+50, y+25, 50*r, 50*r);
39      ellipse(x+25, y+50, 50*r, 50*r);
40  }
```



Generativity

- Rules
- Repetition, but with variation
- Emergence

Example:

<https://rawgit.com/RaggedyAnn/OldMinis/master/5th%20weekly%20mini%20exercise/treeGenerator/index.html>

Rules:

- place one line in the middle of the width $1/10$ of the height long
- translate to the tip of the line
- randomize repeat for either one or two line(s)
- in case of one: repeat
- in case of two: scale, rotate and repeat

Code:

<https://github.com/RaggedyAnn/OldMinis/blob/master/5th%20weekly%20mini%20exercise/treeGenerator/sketch.js>



MiniEx5 – Discussion

Sæt jer sammen i grupper, der ikke er dem I ellers er i gruppe med.

Præsenter jeres miniEx fra sidste uge for hinanden.

- Forklar din vision.
- Vær så præcis som mulig med hvorfor du bruger lige præcis den syntax det sted igennem koden.
- Giv gerne feedback og forslag til hinanden.

MiniEx6

- Make sure you have read the text by Winnie Soon and watched the video lecture by Marius Watz
- Start with a blank paper. Think of at least **three simple rules** that you want to implement in a **generative** program.
- Based on the rules in step 2, then design a generative program that utilizes at least **one for-loop/while-loop** and one **conditional statement**. (You may also consider to use `noise()` and `random()` syntax)
- Upload your sketch to your own Github account under a folder called **mini_ex6**
- Create a readme file (README.md) and upload to the same mini_ex6 directory.
- Provide peer-feedback to 2 of your classmates on their works by creating "issues" on his/her github corresponding repository. Write with the issue title "Feedback on mini_ex(?) by (YOUR FULL NAME)"

readMe

A screenshot of your program

A URL link to your program and run on a browser.

What are the **rules in your generative program** and describe how your program **performs over time**. What have been **generated**?

How does this mini-exercise help you to understand what might be **generativity**?

Peer-feedback

First you describe what is the work, what are the elements in the work? like what you have seen, what you have experienced and what syntax he/she has used.

What is the emphasis? What does the work express? What does this work say or mean to you? How would you interpret the work?

Do you like this program, and Why? and which aspect do you like the most?

Provide suggestion for improvement or expansion of the program/thoughts

3, 2, 1, -CODE!!!