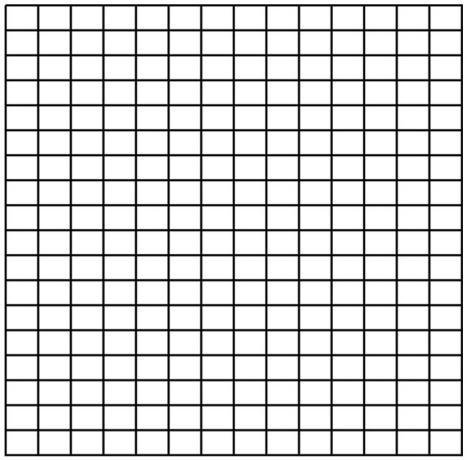
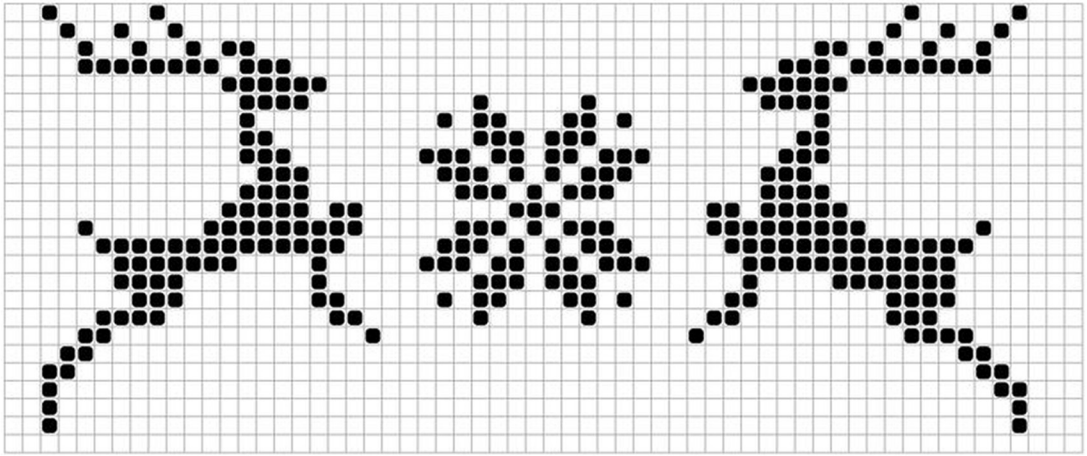


blank knitting chart



knitting pattern on a knitting chart

A knitting pattern is a set of instructions on how to construct items using knitting.

Chart patterns use a matrix of blocks filled with letters and symbols to describe the knitted stitches, typically with one stitch per block.

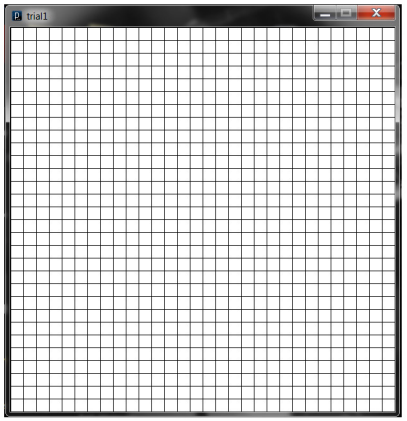
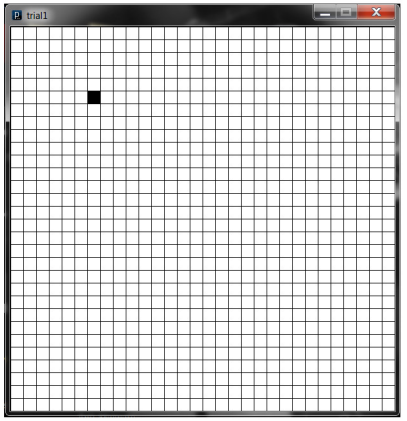
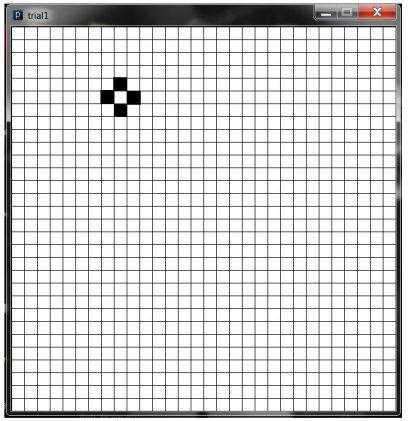
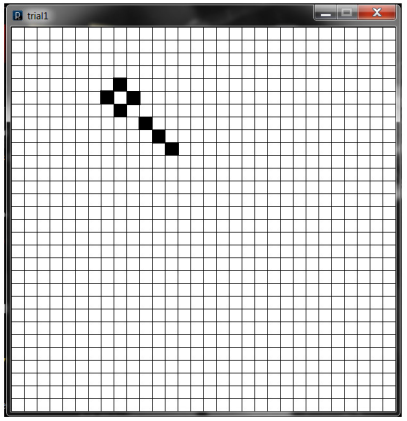
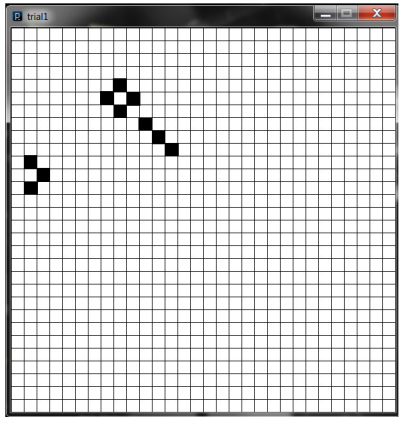
Chart patterns provide visual feedback on the relative position of stitches.

They may be color-coded for multi-color knitting.

(information from wikipedia)

Manual Chart

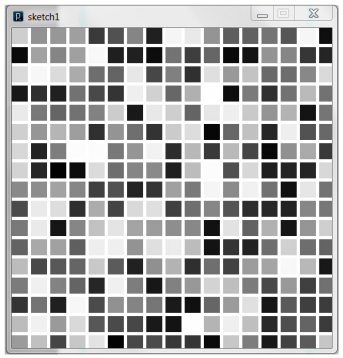
```
int i;
int j;

void setup(){
  size(600,600);
  background(255);
  smooth();
}

void draw() {
  //background(255);
  for( int i = 0; i < width; i = i + 20) {
  line(i, 0, i, height);
  stroke(0);
}
for( int j = 0; j < height; j = j + 20){
  line(0, j, width, j);
  stroke(0);
}

if (mousePressed == true) {
   //line(i, j, mouseX, mouseY);
   //stroke(0);
   rectMode(CORNER);
   rect(round(mouseX),round (mouseY),20,20);
   fill(0);


 }

}
```

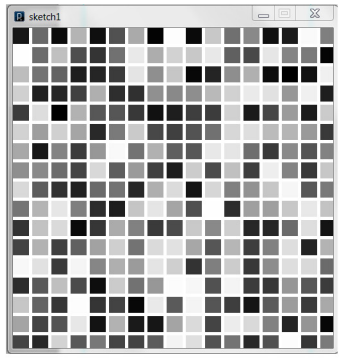






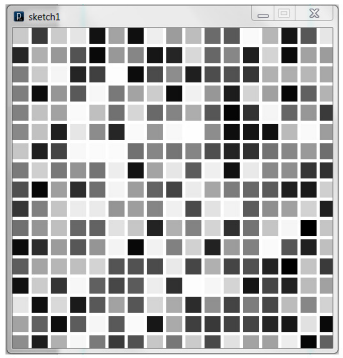
This code is produced for mimicking the knitting charts. It enables filling the chart cells manually in a very simple way. Knitting charts are basically an abstract way to describe the knitting process. It is one important aspect how one kind of information contains physical actions like purl, knit, stitch converted to 2D layout. Taking this issue into consideration, concentrating on knitting patterns and applying a generative characteristics to those charts could be an interesting point of view in terms of computational data of knitting.

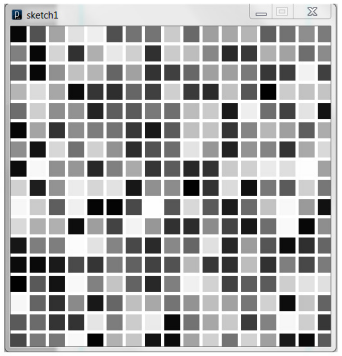
Random Generating Chart

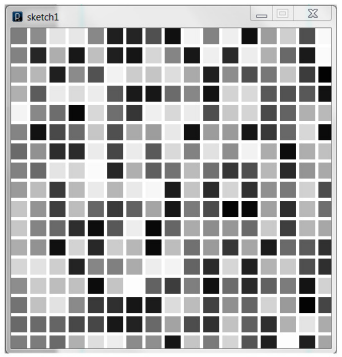
```
void setup(){
size(500,500);
//background(255);
smooth();
}


void draw(){
background(255);
int cols = width;
int rows = height;
int [][] myArray = new int [cols] [rows];

for( int i = 0;  i < cols; i++){

for( int j = 0; j < rows; j++){
    myArray[i][j] = int(random(200));
    //stroke(myArray[i][j]);
    noStroke();
    rect(i*30,j*30,25,25);
    fill(random(255));
  }
 }
}
```







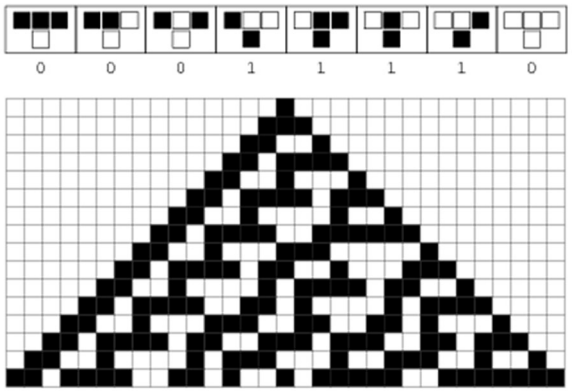




The idea of a generative chart which produces patterns. This patterns contain a mono-chromatical hue which refers each color code presents one kind of operation. The code, at that point, is not following any rule.

As long as it is random, the information has no meaningful computational qualification and been lacking to fulfill the process of knitting properly. In order to make it easy to follow, a set of rules should be defined.

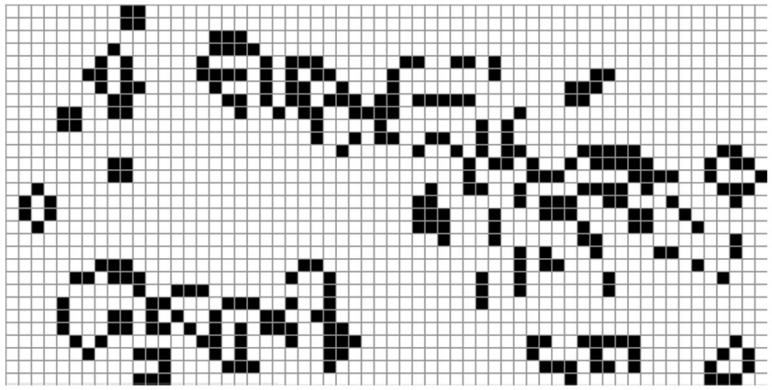
Cellular Automaton.The Game of Life



Cellular Automaton is a set of rules for the cells to follow on a grid. Each square is called a "cell" and each cell has two possible states, black and white. The neighbourhood of a cell is the nearby, usually adjacent cells. The idea was developed by two mathematician, Stanislaw Ulam and John von Neumann in 1940's.

Years later, John Horton Conway, a British mathematician, developed the concept of "Game of Life". The "game" is a zero-player game, meaning that its evolution is determined by this initial state of a cell requiring further input.

One interacts with the Game of Life by creating an initial configuration and observing how it evolves or, for advanced players, by creating patterns with particular properties.



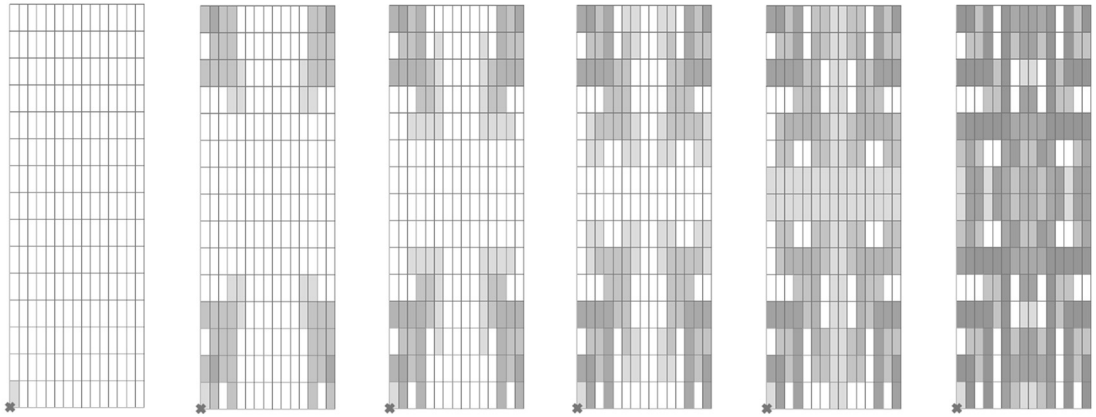
The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead. Every cell interacts with its eigt neigbours, which are the cells that are horizontally, vertically or diagonally adjacent.

At each step in time, the following transitions occur;
*Any live cell fewer than two live neighbours dies, as if caused by under-population.
*Any live cell with two or three live neighbours lives on to the next generation
*Any live cell with more than three live neighbours dies, as if by overcrowding
*Any dead cell with exactly three neighbours becomes a live cell, as if by reproduction.

The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed-births and deaths occur. The rules continue to be applied repeatedly to cerate further generations.

Generative Knitting Chart

Initial Cell State.1 Initial State



initial state

cells are generating as live and dead, system automate itself according to the reproduction and demising rules

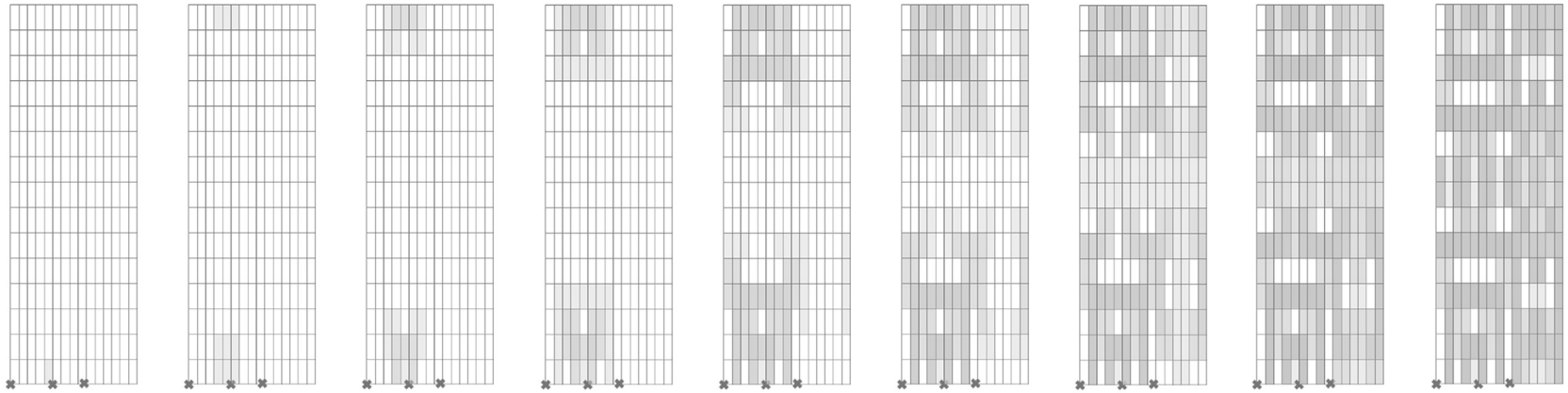
Cellular automaton logic, as a generative system, is applied to the 2D knitting chart. Every square in the knitting chart behaves according to the initial state and cellular automaton rules.

The monochromatic hue depends how many times the cells were reproduced; more darker cells present more than one reproduction.

Different patterns occured depending on the position and amount of the initial states. Therefore, the parameters can be set as;

*Position of initial state
*Amount of initial state

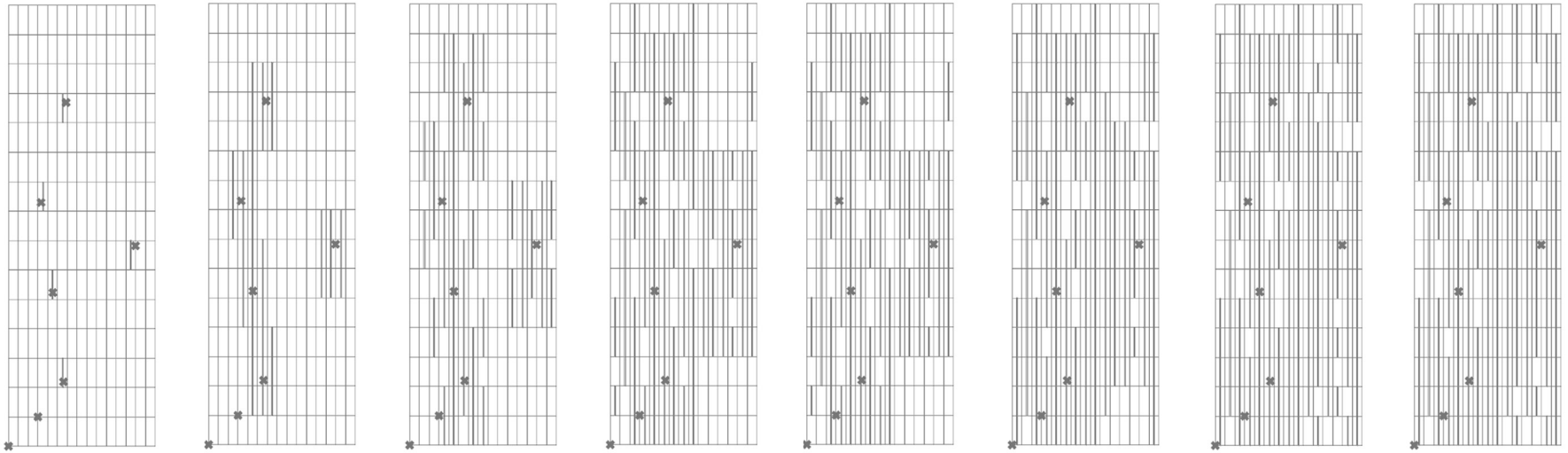
Initial Cell State.Multiple Initial States



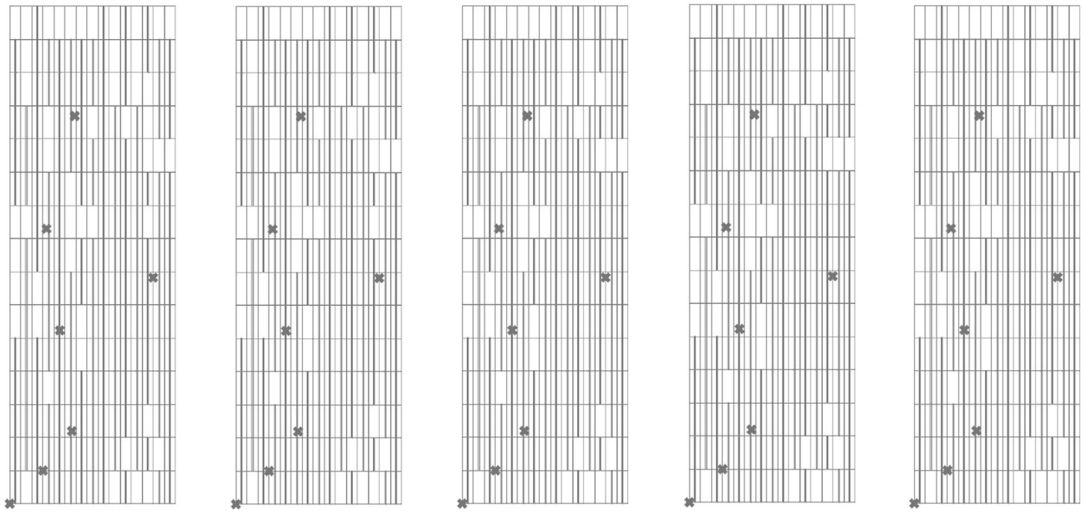
initial states

Generative Knitting Chart

Multiple Initial States.Different Positions.Stitches



initial states



These chart series contains the data of the amount and the position of the stitches.

The process of knitting is formed by a generative system, cellular automaton. The chracter of the knitting pattern depends on the production process which is evolved the feature of initial state (IS). The amount and the position of the IS are the parameters besides the cellular automaton parameters.

In conclusion, the aim is to enhance the information from knitting chart and apply a computational aspect to the process.

Generative Knitting Chart.Code

Grasshopper Definition