DATT 2040

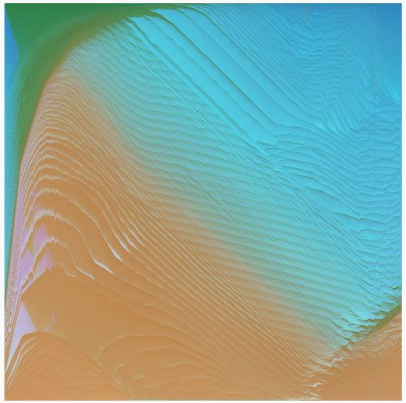
Math Code Art

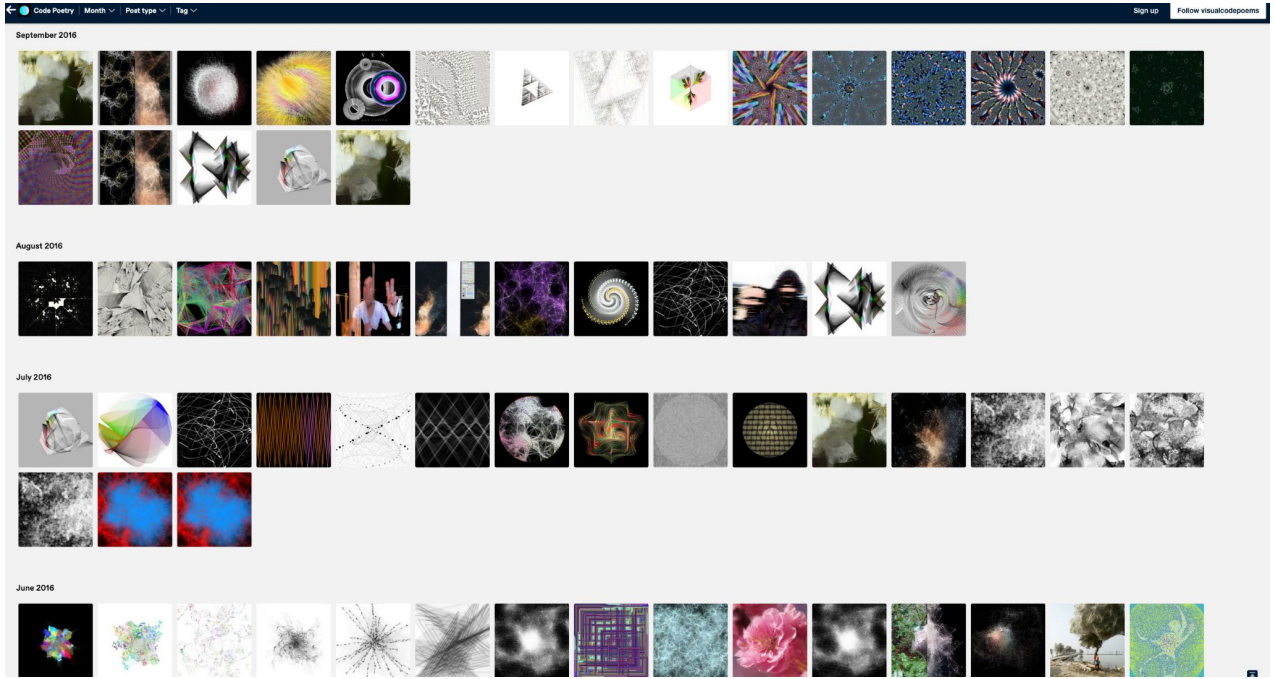
About me

Home Archive

Code Poetry

Code artworks, inspired by zen poetry and haiku. Celebrating the simplicity and beauty that can be found in code.
Updated daily. More work can be found at www.dantappersounddesign.com
Contact: dantappersounddesign@gmail.com





https://visualcodepoems.tumblr.com/

Math Art Code . DATT 2040

Course Director: Dan Tapper <dantap@yorku.ca>

Course Github repository: https://github.com/atarilover123/DATT-2040-Math-Art-Code

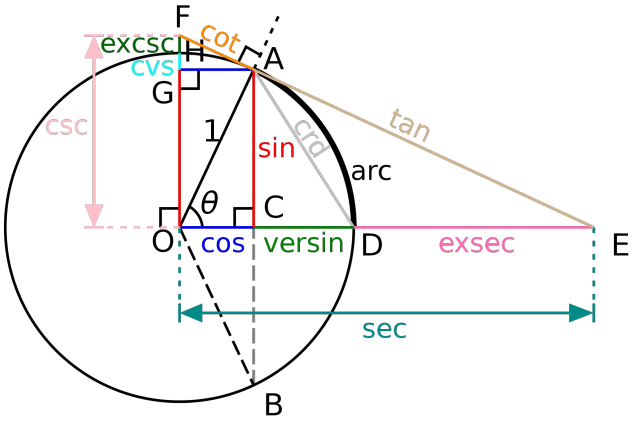
Course calendar description

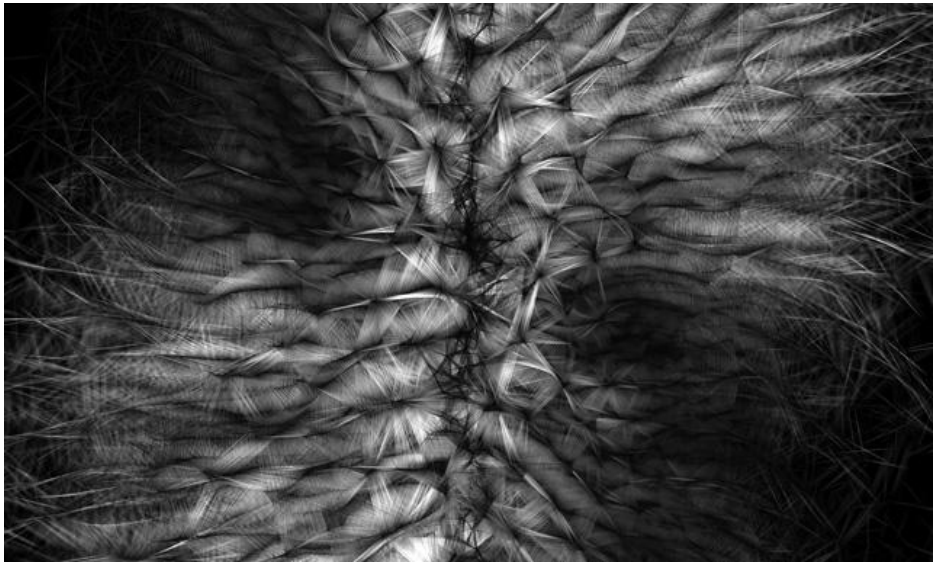
Math Art Code explores relationships at the intersection of math, art and creative coding by providing a survey of mathematical concepts and techniques that support creation processes in contemporary code-based art and performance practices. Through a series of weekly studies reviewing mathematical concepts, their programmatic applications and investigations into the artists who employ them, we will build our own creative coding toolset. This toolset can be applied to the creation of computational artworks, such as video games, mixed reality, digital fabrication, and image manipulation and will provide a pathway from simple computational sketches to more advanced projects.

This course will be delivered through a combination of:

Weekly lectures – exploring artists using mathematical concepts, looking deeply into computational artists and investigating their use of code, math and algorithm to create a wide range of computational and multimedia artworks. **Lectures will be held in CC 211.** Slides will be posted online in the Course Github repository and on Eclass.

Weekly labs – Learning coding and mathematical concepts and putting these into practice – creating our own generative art works and computational media projects.
Labs will be held in ACW 102.





Required tools

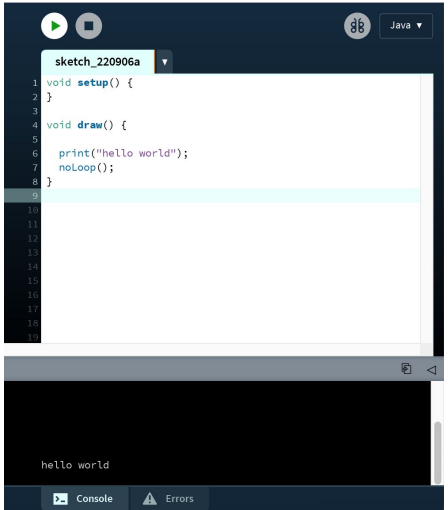
The practical element of this course will be taught using a modular suite of tools which have been selected to showcase the versatility of creative coding practices.

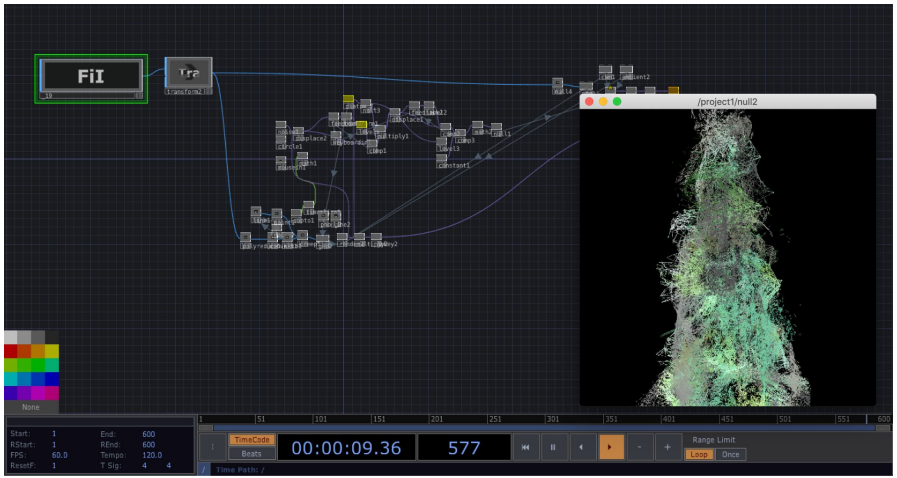
We will be using Processing/p5.js as our primary development tool and exploring TouchDesigner and MaxMSP to expand knowledge of platforms and engage with audio processing, shaders and 3d rendering environments.

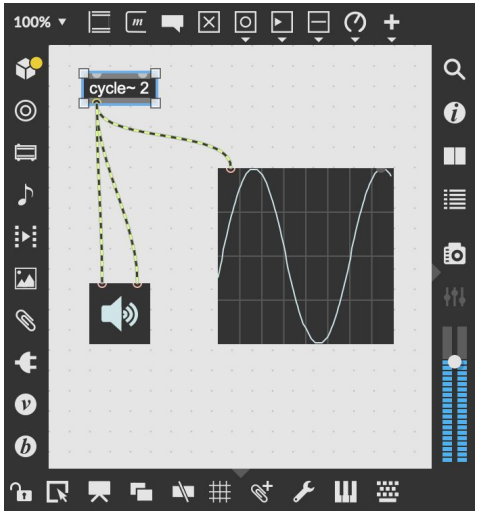
Processing – https://processing.org/

TouchDesigner – https://derivative.ca/

MaxMSP – https://cycling74.com/







Welcome to Processing!

Processing is a flexible software sketchbook and a language for learning how to code. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

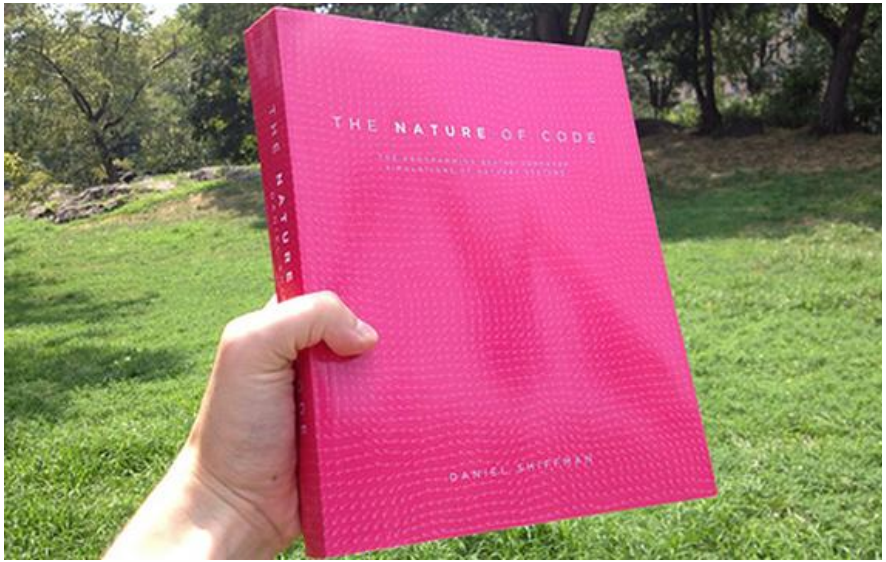
Download | Reference | Donate

Submissions and class resources

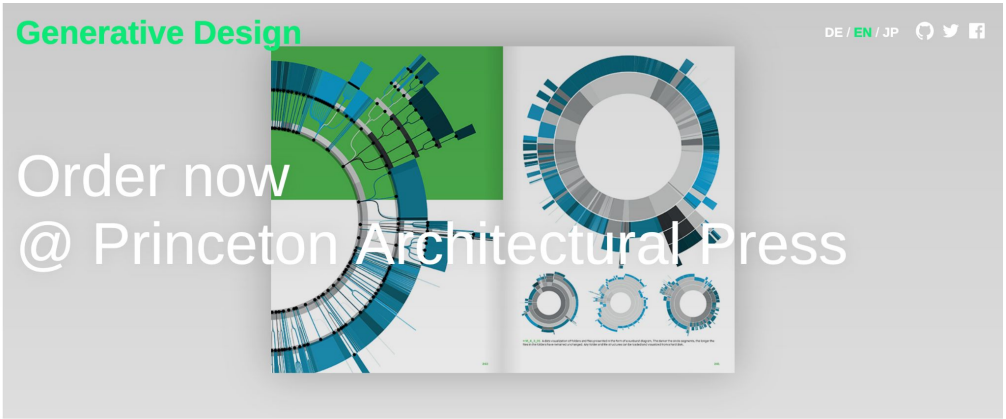
Students will create a Github account to upload weekly sketches and notes. Tests and quizzes will be conducted via Eclass.

Github – https://github.com/

eclass – https://www.yorku.ca/eclass/



[Nature of Code by Daniel Shiffman]



Hello and welcome to Generative Design, Creative Coding on the Web. Here, you will find all of the sketches from the book and their associated code. Run

[Generative Design by Benedikt Groß]

Topics and concepts

Through exploring a number of computational and conceptual artists we will explore how topics such as modular arithmetic, number systems, boolean algebra, combinatorics, vectors, trigonometry, coordinate systems, linear algebra are used in the creation of generative artworks such as **video games**, **mixed reality**, **digital fabrication**, **audio creation** and **image manipulation**.

Learning outcomes with examples

Students will be able to:

- Think critically about digital artworks and have an understanding of the underlying algorithms used to create them.
- Have an understanding of mathematical concepts and be able to apply these concepts to the creative process.
- Think modularly about problems and tools, utilizing knowledge of fundamental math and code concepts to work in multiple environments.
- Create their own computational artworks using math and code as a creative medium.

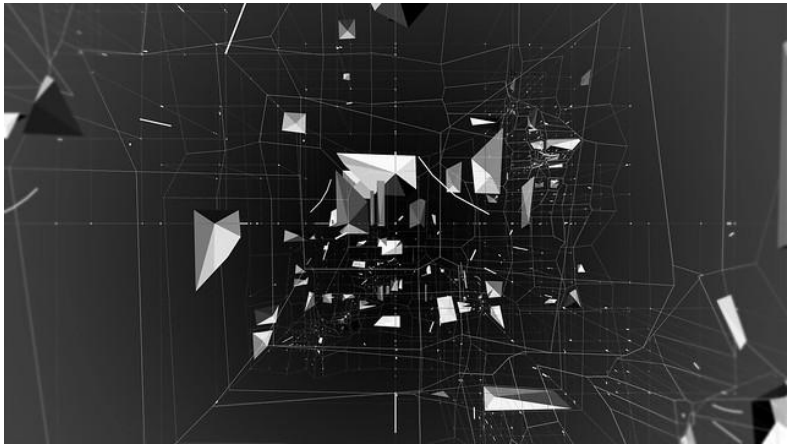
Module breakdown

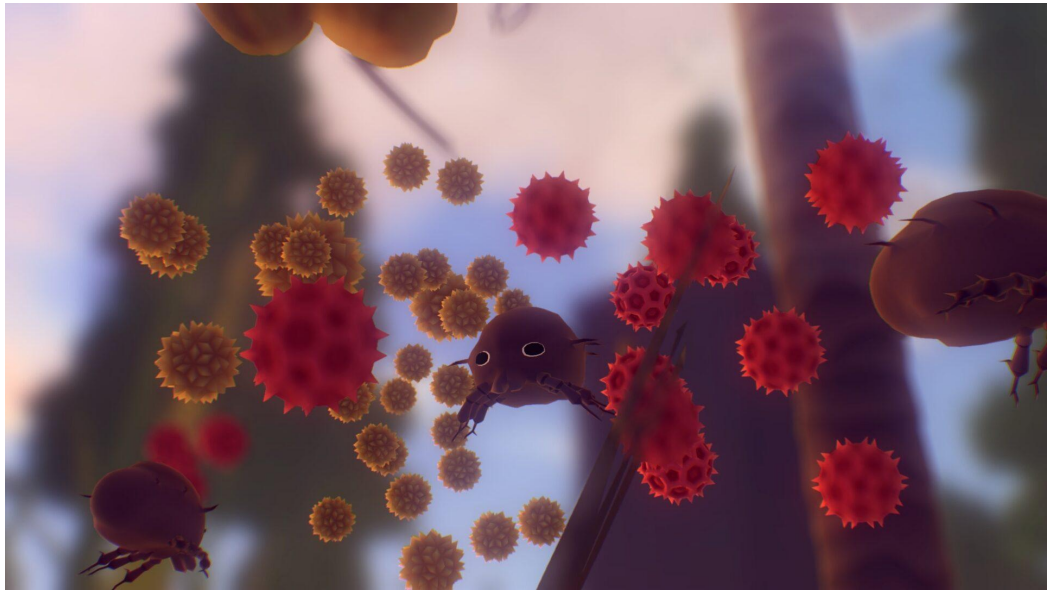
Weeks 1 - 4, Generative art

Weeks 5 - 6, Natural algorithms and feedback

Weeks 7 - 9, Multimedia processing

Weeks 9 -12, 3D world building and final projects





Graded assessment

- Sketches: 30%
- Quizzes: 20%
- Midterm exam: 15%
- Final exam: 15%
- Final creative project & evaluative essay: 20%

Sketches are regular creative assignments that apply the mathematical and artistic concepts from class to the creation of a computational arts-based study.

Sketches will be marked out of **5 points** and **experimentation is encouraged.**

Each week we will begin work on a sketch in lab, it is ok to submit work that utilises and iterates on the code we learn as a group in lab.

- Submission of a sketch using code from lab with no augmentations or artistic modifications: **3 points**

- Submission of a sketch that uses code from lab but develops on this code to create more unique artworks - think adding new variables, altering the core algorithm, adding parameterised control or all of the above: **4 points**

- Submission of a sketch that significantly goes beyond what we look at in class, this could involve significant additions to the core code, rethinking the mathematical/code approaches and programming the core algorithm in a different way (there are always multiple approaches), utilising the core algorithm as part of a larger artwork: **5 points**

Quizzes will focus on artists reviewed in class, mathematical and coding problems and the theoretical concepts associated with them.

Quiz schedule to come by end of week.

Midterm exam will be a comprehensive exam that includes topics from the first half of the course.

Final exam will be a comprehensive exam that includes topics from throughout the course and will include both a mathematical portion with problem solving, and a reflection component on the concepts and art covered in the course.

Final creative project and evaluative essay is a project where students develop a computational artwork combining 3 or more mathematical/coding concepts from the course. This project will be combined with a short evaluative essay explaining the creation process, concepts used and why they were chosen.

Reasons to come to lecture

Reasons to come to lab

Office hours

Week 1

Generative art.1

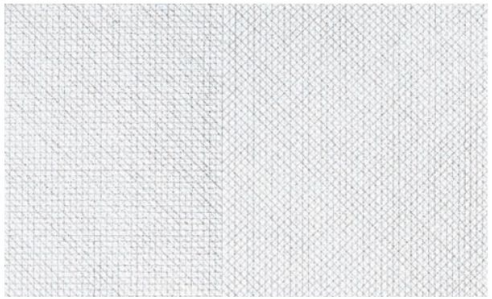
What is generative art?

Algorithm as instruction

Generative art is art made using a predetermined system that often includes an element of chance – is usually applied to computer based art – Tate Modern

The Idea Becomes a Machine that Makes the Art – Sol Lewitt

SOL LEWITT: A WALL DRAWING RETROSPECTIVE

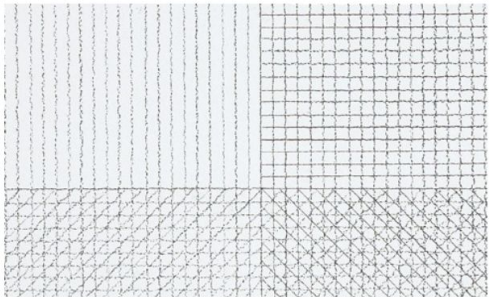


Wall Drawing 47

A wall divided into fifteen equal parts, each with a different line direction, and all combinations. June 1970 Black pen... Read More

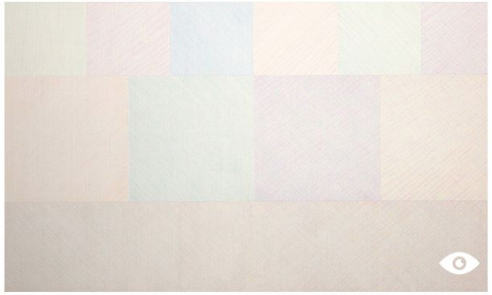
Wall Drawing 51

All architectural points connected by straight lines. June 1970 Blue snap lines LeWitt Collection, Chester, Connecticut ... Read More

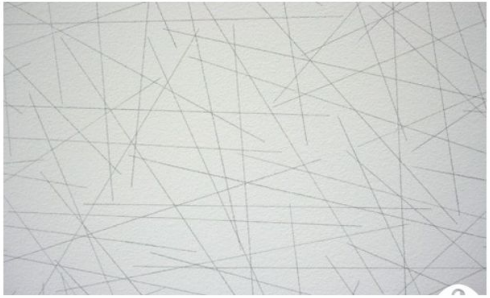


Wall Drawing 56

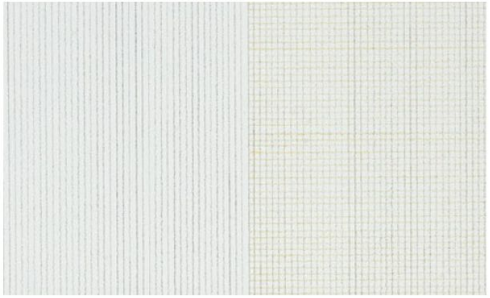
A square is divided horizontally and vertically into four equal parts, each with lines in four directions superimposed p... Read More



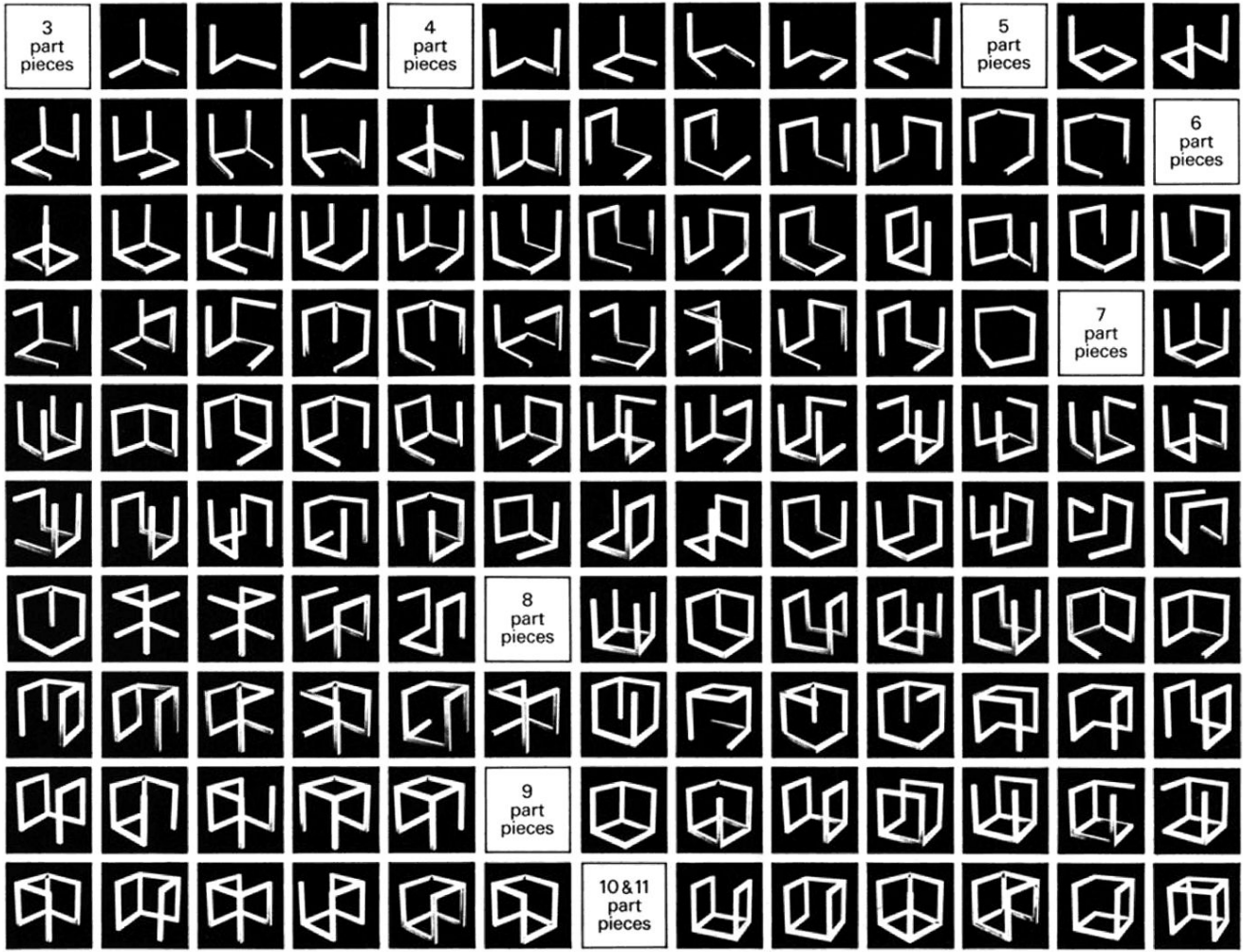
Wall Drawing 85

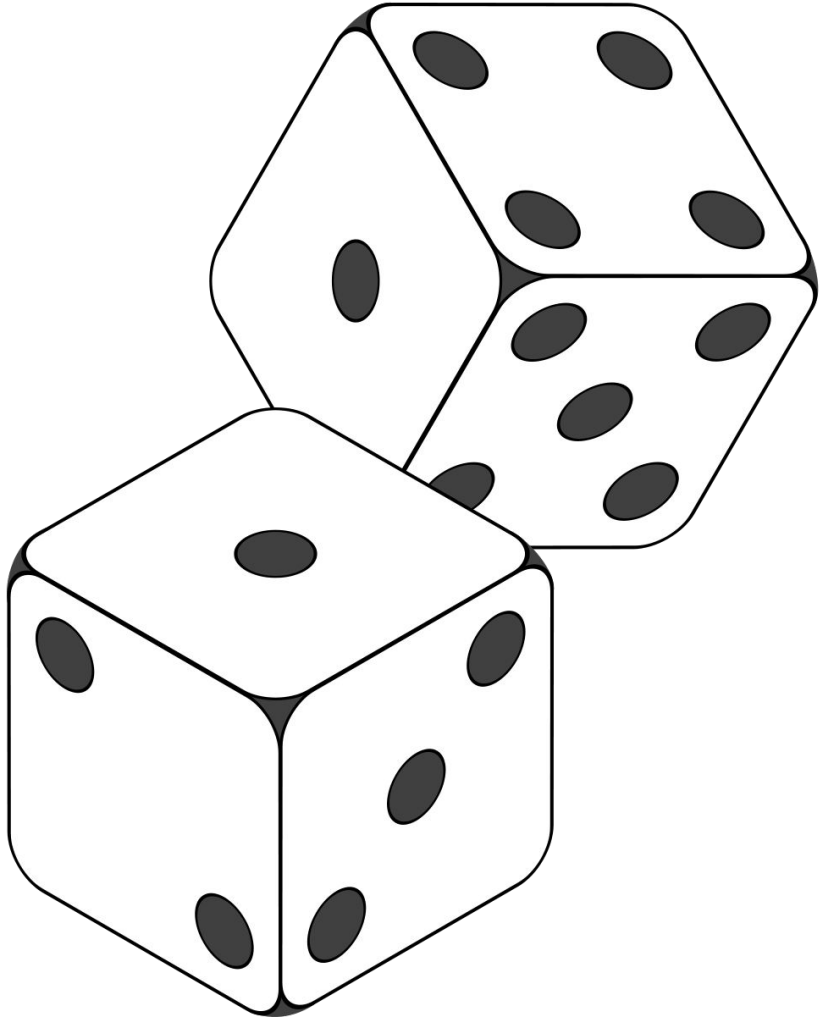


Wall Drawing 86



Wall Drawing 87


3 part pieces
4 part pieces
5 part pieces
6 part pieces
7 part pieces
8 part pieces
9 part pieces
10 & 11 part pieces



1 =
2 =
3 =
4 =
5 =
6 =

Kings (game)

From Wikipedia, the free encyclopedia

For the Japanese party game, see King game. For the Danish film, see King's Game. For the video game company, see King (company).

Kings (also known as **king's cup**, **donut**, **circle of death** or **ring of fire**) is a drinking game that uses playing cards. The player must drink and dispense drinks based on cards drawn. Each card has a rule that is predetermined before the game starts. Often groups establish house rules with their own variation of rules.

Contents [hide]

Equipment [edit]

- 1 Deck of Cards
- 2 or more players
- Alcoholic beverages – typically wine, beer, or mixed drinks - or non alcoholic beverages
- A large cup which will be used as the King's Cup, or (in the "Ring of Fire" version of the game) an unopened beer can

Kings

Cards during a game of Kings

Alternative names	King's cup Donut Circle of death Ring of Fire Four Kings
Type	Drinking
Players	2+
Age range	Varies by legal jurisdiction
Cards	52
Deck	1 deck of standard playing cards
Play	Clockwise or counterclockwise

THE ALGORISTS by Roman Verostko

Jean-Pierre Hébert

Chicago, 1992,
sienna natural and
cobalt blue;
pen and inks on
technical paper.

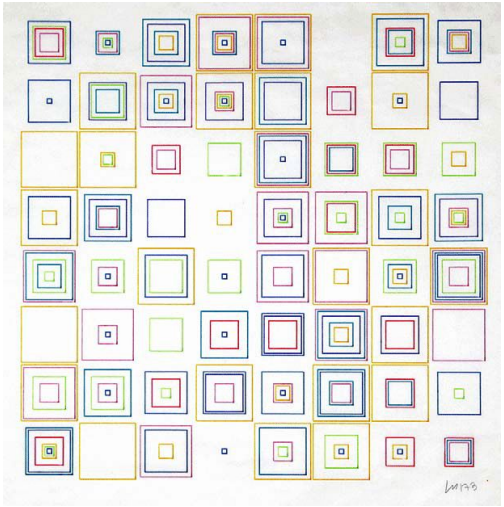
© j.p.hebert

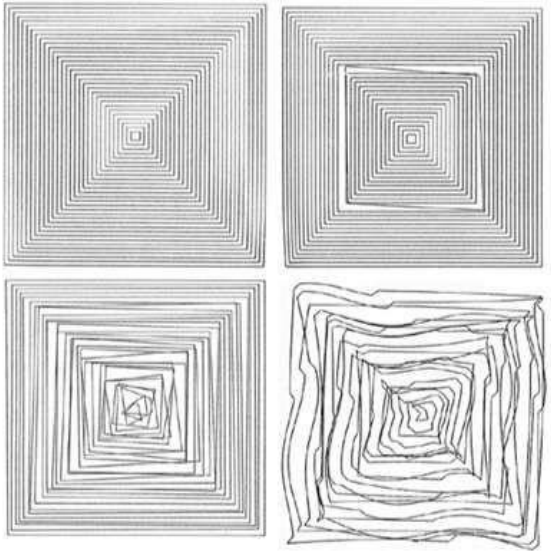
Often I am asked "Who are the algorists?" Simply put an algorist is anyone who works with algorithms. Historically we have viewed algorists as mathematicians. But it also applies to artists who create art using algorithmic procedures that include their own algorithms. This page presents an account of the adaptation of this term by a group of artists in 1995. Algorithmic art has a deep history that reaches back to prehistoric art. But the advent of computing power spawned an artistic practice with form-generating features that is relatively unique to the last quarter of the 20th Century. Computer power gives the artist's algorithms a leverage we might liken to the power of the engine in the industrial revolution. In 1969, when I first tasted algorithmic leverage, I set out to learn how to use it as an artist.

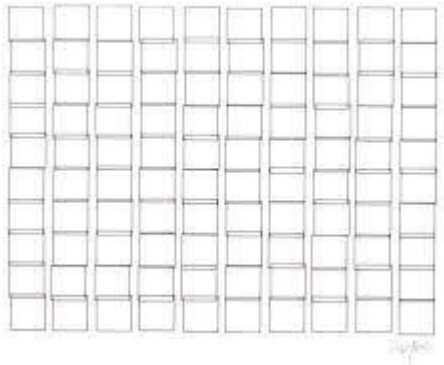
veramolnar.com

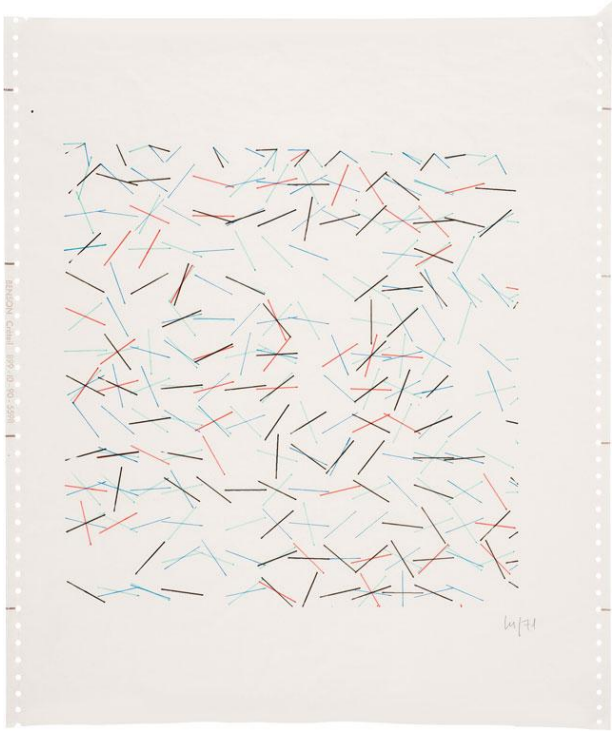
Vera Molnár is a Hungarian media artist living and working in France. Molnar is widely considered to be a pioneer of computer art and generative art, and is also one of the first women to use computers in her art practice. Born in Hungary, she studied aesthetics and art history at the Budapest College of Fine Arts. Wikipedia

Born: January 5, 1924 (age 98 years), Budapest, Hungary



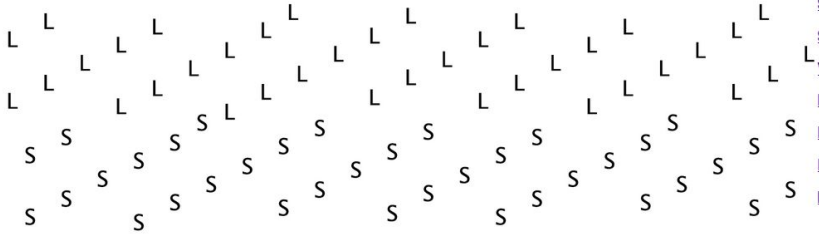


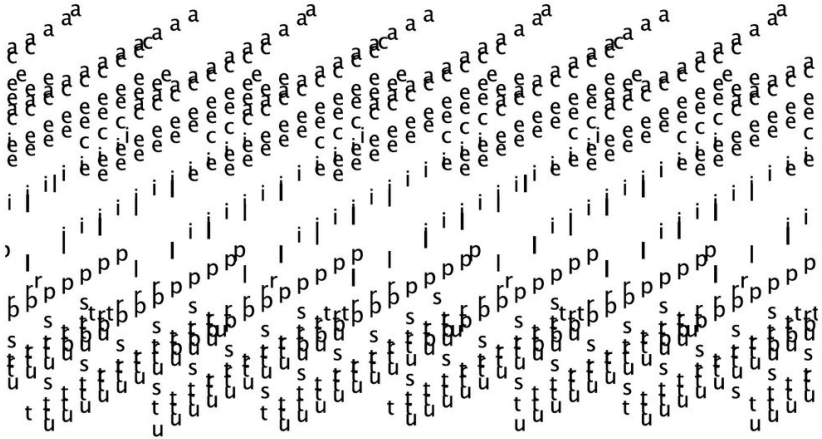


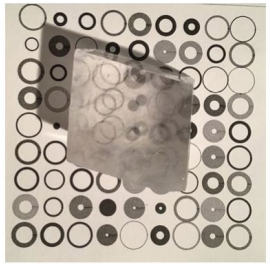


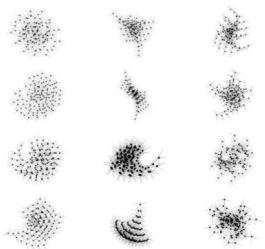
SuperLattice

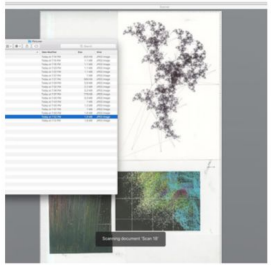
A visual guide to the components of SuperLattice as a theoretical exhibition











Grids

In our first lab we will be thinking about grids.

- How can we draw a grid through code?
- How can we make a simple and highly defined structure more interesting?
- How far can we deconstruct a grid structure before it stops resembling a grid?

To lab bring.

- Yourself
- Pen and paper

Reading to do before lab (at the very least skim through and look at the pictures):
https://www.rightclicksave.com/article/an-interview-with-vera-molnar

And this Processing overview