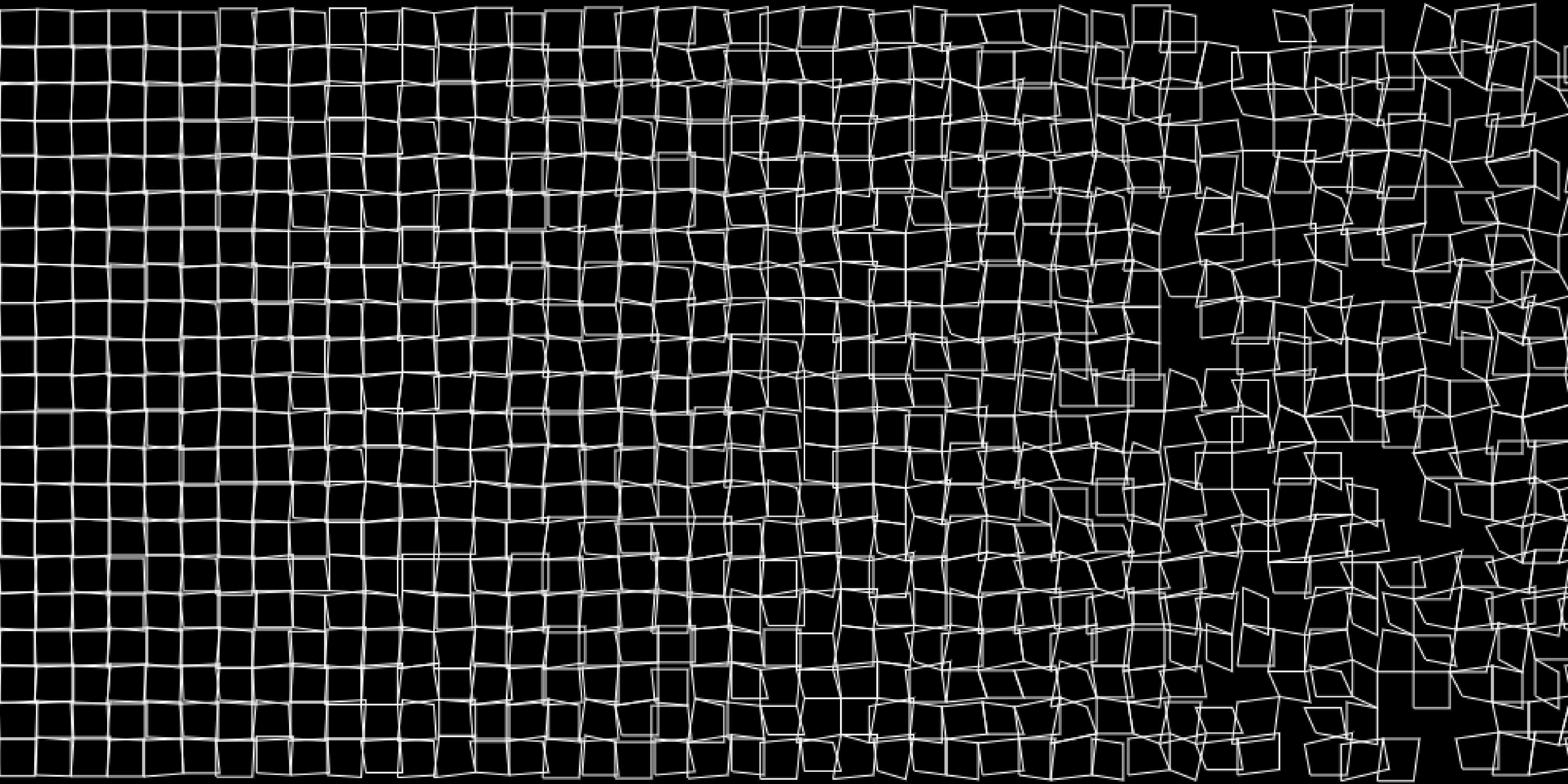
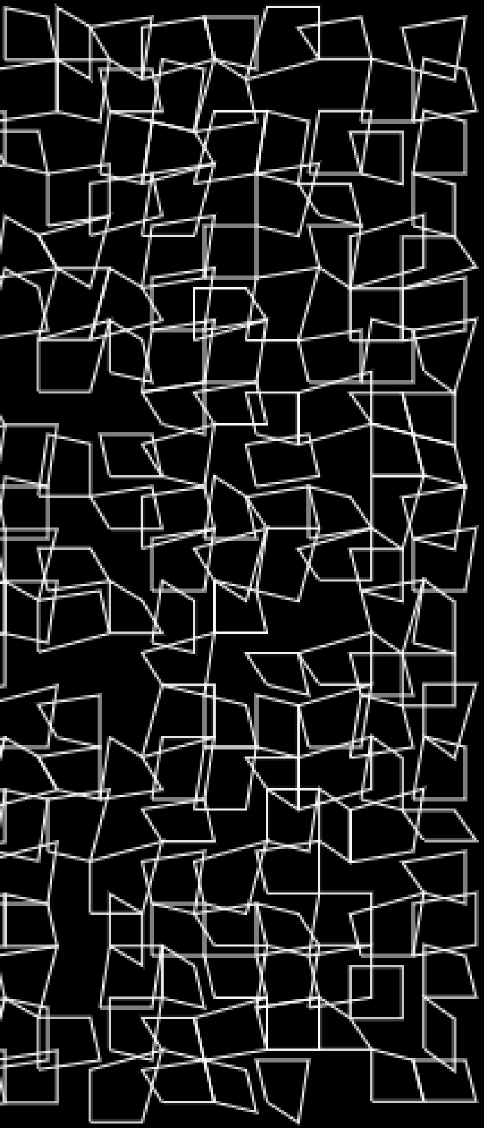




ARTIFICIAL CREATIVITY

Bachelor Thesis in
Visual Communication
by Bjørnar Øvstedal
2022

Designer

Bjørnar Øvstedal

Edition

Bachelor Thesis BAV3
2022

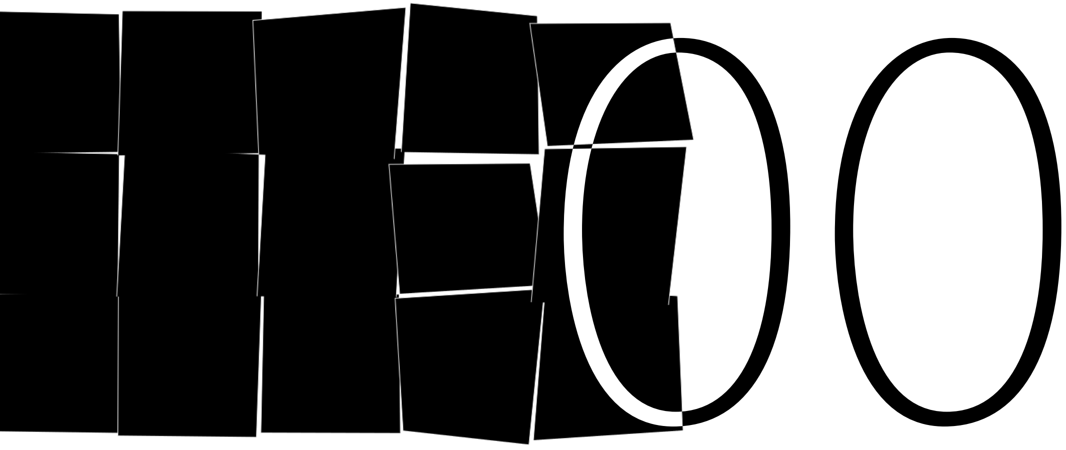
Institution

Bachelor Visual Communication

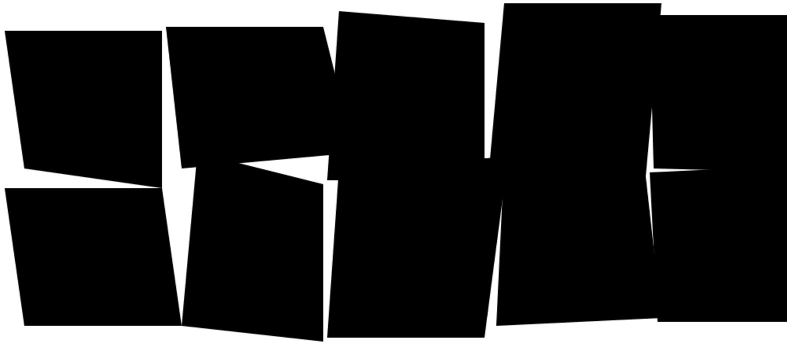
University of Bergen
Faculty for Art, Music and Design (KMD)
Institute of Design
Norway

—

Typeset in;
Titles: Kozuka Gothic Pro
Body text: Scala Sans


00

Foreword



> *“The intolerable labour and fatiguing monotony of continued repetition of similar arithmetical calculations, first excited the desire, and afterwards suggested the idea, of a machine, which, by the aid of gravity or any other moving power, should become a substitute for one of the lowest operations of human intellect.”*
>
> *— Charles Babbage (1822)*

Computational thinking is not a new concept. Rather, it sprung into existence millennia ago with the ancient Babylonians and Mesopotamians. Arithmetics, as a tool, was useful in their everyday lives to calculate the plot of a land among many other things. Historians have concluded that the Pythagorean theorem was in widespread use not only in state of Babylonia, but also in Sumer. *Computation*, i.e. a calculation or a series of calculations that follow given steps within a well defined model, has been precent among humans long before the invention of the computer. However, today, the terms computation and computer have become entangled with digital computation.

But computers are not magic, and neither are the algorithms they run. Increasingly arcane, perhaps, but not magic. They are still a series of steps so well-defined any actor can follow them; even actors are stupid as machines.

A Step Back

It is no secret that machines have become exponentially more advanced, and with the recent progress of AI, the halting of that exponential advancement remains unlikely. It is becoming harder for humans to maintain an overview of the series of steps being executed. The output can now only be partially predicted or, in some cases, not at all.

My Stake and Motivation

I continue to be fascinated by computers, and have on several circumstances sought to integrate computational and information theory into my endeavors within the field of design. In my opinion there exists a seemingly endless well of tools to boost aesthetics, some we have taken advantage of and some we have yet to. It remains paramount to map these to the best of our abilities, uncover the pitfalls we might fall into along the way, and, perhaps, spend some time reflecting on the philosopical questions that inevitably arise.

01

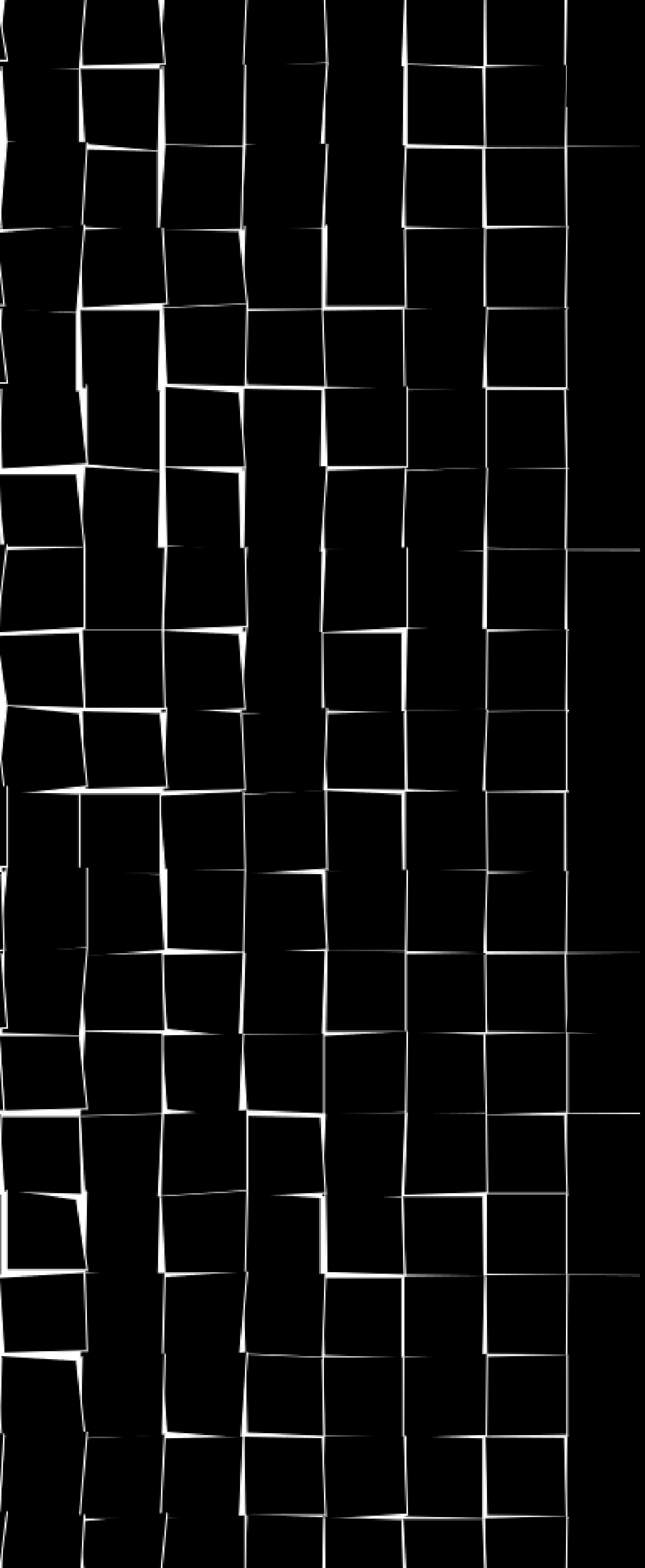
Introduction

This bachelor project concerns computer science and its use cases in the field of design, while analyzing the appliance of artificial intelligence — a technology still in its infancy — in aesthetics fields. Ethical, legal, and societal quandries continue to surface; morality may need to be meticulously defined, copyright laws need to be revised, and the displacement of workers need to be addressed. All these questions and likely many, many more face humanity going forward. In my thesis I will address some of those that relate to the field of aesthetics and the discipline of design; a broad term, encompassing many aspects and potential problems.

It is my intention, by shedding light on some of these issues, to encourage introspection and speculation, as opposed to providing solutions for these questions, an attempt which would be foolhardy indeed granted the difficulty many of them pose. Mainly, I seek to explore how *computational thinking* challenges our perception of *creativity*. I will use algorithms and computation as a primary method of designing visuals, while relying on a pen plotter to translate the output onto a physical format.

Computational creativity is a field of research with much attention and many subfields; musical creativity, linguistic creativity, visual and aesthetic creativity, and the creativity tied to problem solving.

Creativity is a faculty of intellect, and thusly a subdomain of artificial intelligence. In this thesis I will discuss the metrics to evaluate creativity and if it is at all possible for machines to fulfill them.



Research Question:

How can generative design challenge our perception of creativity?

02

Boundaries

Terminology

In order to have a fruitful discussion about the topic of artificial creativity, it is important to share a framework of definitions. Generative design is historically a new term and thus perhaps not in the common vocabulary. Creativity, on the other hand, rests diametrically opposed; it is an old concept with many competing definitions.

Generative Design

Generative design is commonly thought of as a tool for architectural design or engineering; instead of relying on the computations of a human to determine the optimal parameters for load bearing, aerodynamics, space optimization, or plasticity, one can employ software to handle these streinous calculations.

When considering generative design in this thesis it is in a broad sense, not as a tool to solve practical problems. While I at times will wrestle with this as well, particularly in the section on analysis, I will more so consider the aesthetic and technical value — primarily those relating to expedition — of generative design approaches.

For these reasons I will make no distinction between generative 'art' and 'design', and I will use these terms interchangably.

The Metrics to Measure Creativity

While there exists many definitions, for our intended purpose the one of Margaret A. Boden, a Research Professor of Cognitive Science in Sussex, is most apt. In her book 'The Creative Mind' she outlines a set of attributes for which to evaluate the act of creativity. According to Boden, creativity is the ability to come up with ideas or artefacts that are *new, surprising*, and *valuable*.

Novelty
An idea can be new either in a historical sense or in a psychological sense. These are divided into H-creativity and P-creativity respectively, whereas H-creativity is a subcategory of P-creativity (i.e. all H-creativity is also P-creativity). An idea is H-creative if no-one previously in all of known history has not conceived of the idea prior. P-creativity are ideas new to an individual.

Seldomly would we rob P-creativity of its value. The conception of the same metaphors Shakespear utilized centuries ago in his plays would still be lauded as a creative act, so long as the conceiver had little to no knowledge of Shakespear's body of work. In many ways we would elevate this creative act above others simply because writing on the level of Shakespear would be a monumental feat.

The Element of Surprise

Something surprising is something that defies expectation. To defy odds is to defy the likely outcome. Winning the lottery, or beating the favorite as an underdog in a race would both be examples of this, though perhaps not equally unlikely. Another sort of surprise are ideas that unexpectedly fit into a style of thinking. When you realize it you recognize that it must and always has fit, but no had just discovered it yet. The third form of surprise comes in the form of impossible ideas; something you had not thought conceivable until it was conceived. Ideas so earthshaking that the sky might come tumbling down tomorrow, for now anything might be possible.

Value

And then, lastly, there is the concept of value. This can be economical, societal, aesthetic, political, medicinal, or any other metric you might think of to determine the worth of something. It is also very much subjective, and when in the cases where it *is* objective, it is because one or several of these metrics have been deemed more important than others. One can say da Vinci's *Salvator Mundi* is the most valuable piece of artwork because it sold at an auction for the highest price ($475 000 000; more than $100 000 000 more than the second most expensive art piece ever), but few would agree price is the be-all-end-all of determining value.

And then there are yet three more kinds of creative endeavors; *exploratory creativity*, where an artist explores the fringes of the existing framework, *combinatorial creativity*, where the artist combines two concepts previously thought unreleated, and finally, *transformational creativity,* where the artist conceives of something that fits into neither of the previous two categories. The last, of course, is the rarest, most surprising, and most difficult to achieve form of creativity.

A creative act must satisfy all of the aforementioned conditions to be considered creative. However, to what extent remains to be discussed.

Thematic Limitations

As touched upon earlier, the realm of artificial intelligence and computational thinking is vast; too vast to be considered in its entirety within one bachelor thesis. As such, certain limitations must be established.

Firstly, this thesis will primarily tackle the philosophical questions related to that of creativity when exhibited, or not exhibited, by machines or software. Furthermore, though the line between art and design remains blurry, I will consider artistry both in the form of visual communication and in the broader sense.

Technical Limitations

Hardware

The hardware will be limited to one pen plotter, and likewise the ultimate result to be exhibited will be constrained by this expression. Building machinery requires both time and money, of which I have a finite amount to spend. Additionally, this is not an engineering project, so I intend to keep this aspect to its minimum.

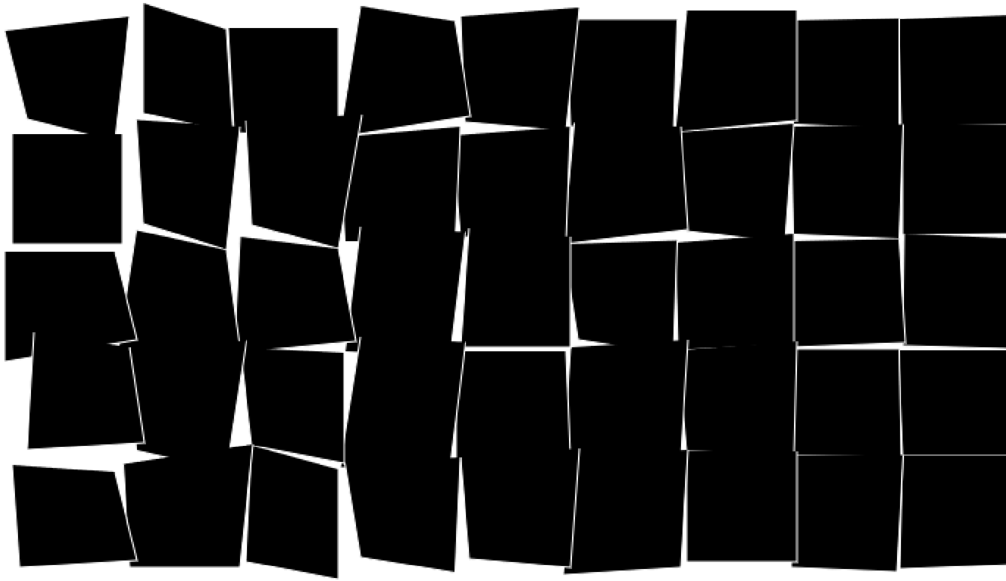
Because I intend to transpose most, if not all, of my digitally generated visuals onto paper using the pen plotter, I have to abandon one of the intrinsic strengths of generative design; animation. Since a computer is able to render many different iterations with varying parameters, it is simple to create motion.

Additonally, the pen plotter limits me to simple colors; gradients and multiple colors can be hacked, but only through considerable effort.

Software

The software to produce visuals will be limited to html, css, and javascript. This is because I am extensively familiar with these languages, recently more so than any other, and I believe these are best suited for non-intensive visual output. It is fast and easy to render to the canvas, and it depends on nothing but a web browser to function, applications present on nearly every modern computer.

P5.js is a tool running in the web browser specifically for visual programming which builds on javascript which simplifies certain parts of the process. While I am familiar with it, I find more flexibility when sticking to vanilla as opposed to relying on pre-implemented functionality. However, there are certain features that could still prove useful, mainly its noise algorithms.



03

Analysis

A Brief History of Generative Art

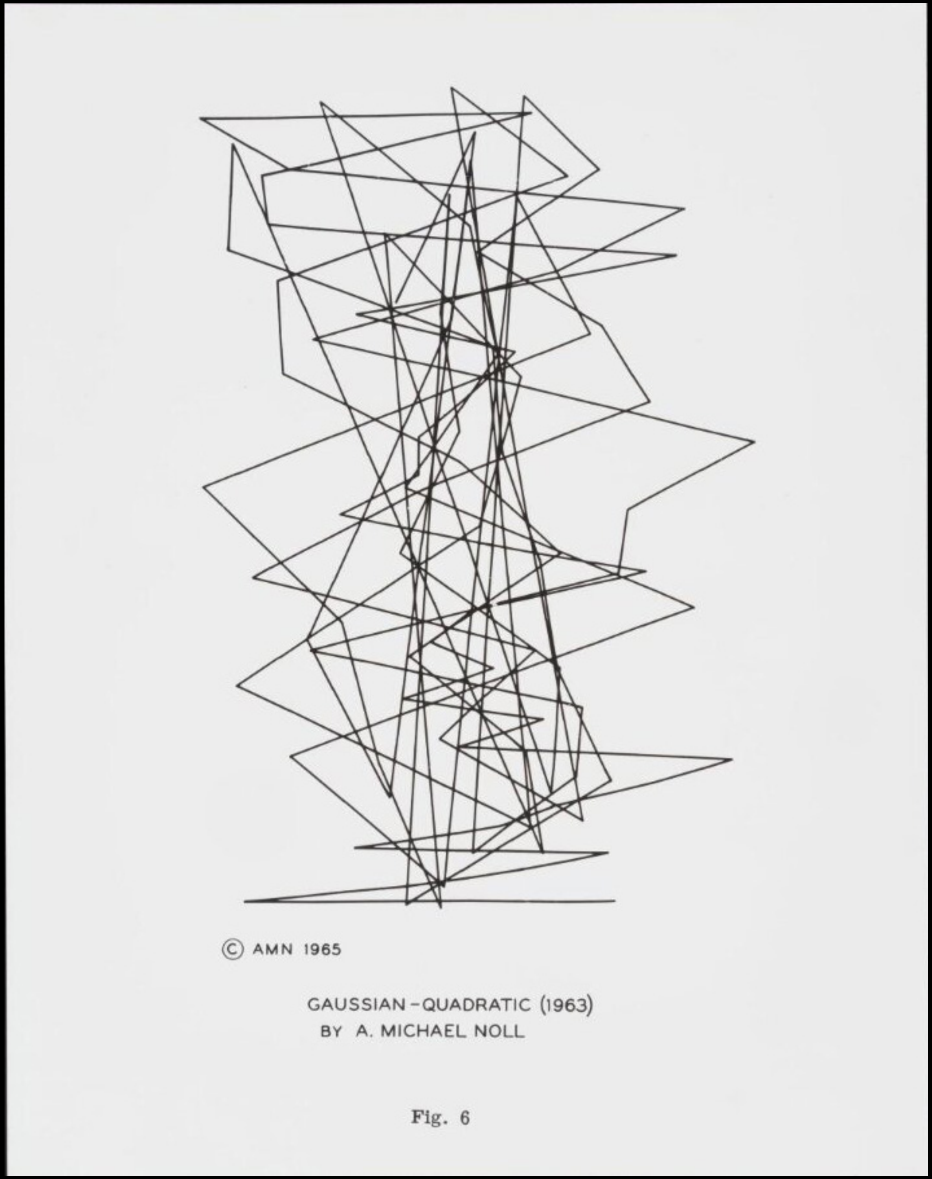
While the term ‘generative’ in reference to art and design remained unsolidified until the 90s, the practice traces back to as early as the 1960s. Georg Nees and Frieder Nake exhibited artwork produced by computers in 1965, which they then titled ‘computer-grafik’.

The Stylistic Simplicity of the 60s

Computer science as we know it today was still being developed in the decades immediately following the war, and many universities were starting to establish their own branch of computer sciences in the 60s and 70s. The tools available to artists were severely limited in processing power, so the output of the era might look simplistic compared to the hyper-realistic renderings computers are capable of churning out today.

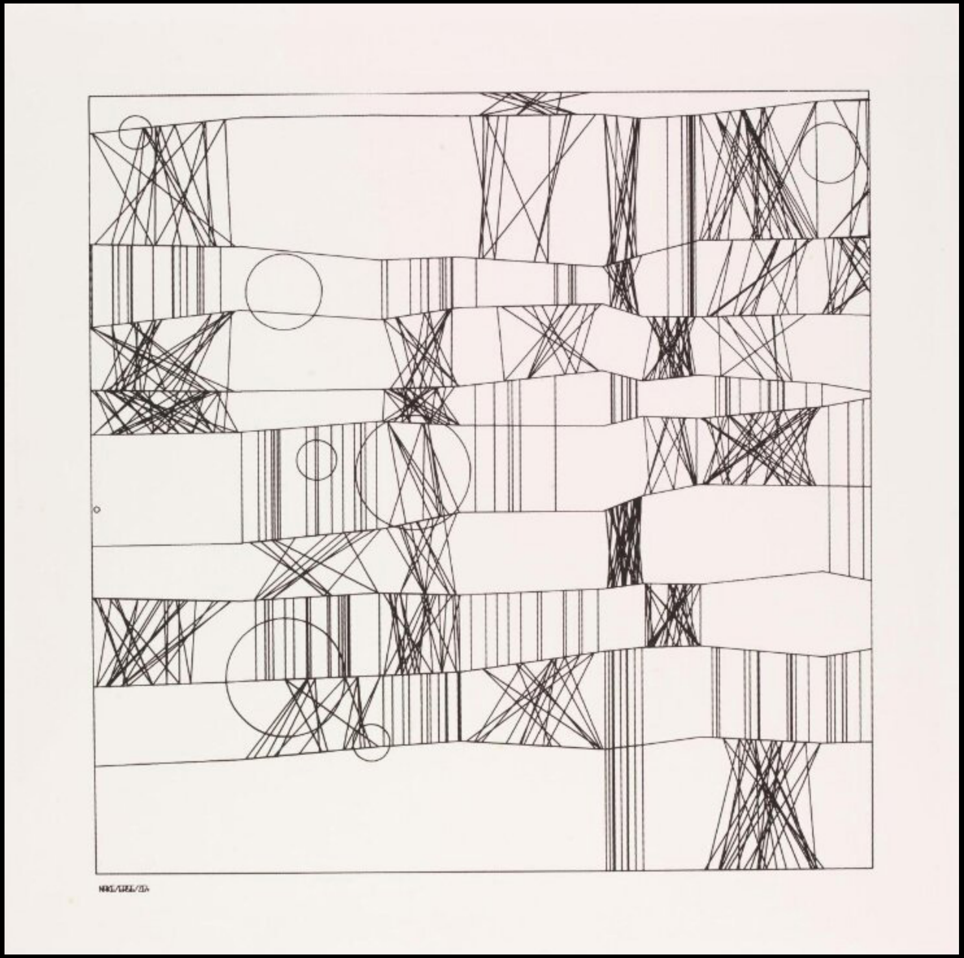
However, there were some who recognized the potential and sought to experiment with this new technology. George Nees, Frieder Nake, A. Michael Noll, and Vera Molnàr are considered pioneers of digital computer art, and I will look at some of their work.

> *“In the computer, man has created not just an inanimate tool but an intellectual and active creative partner that, when fully exploited, could be used to produce wholly new art forms and possibly new aesthetic experiences.”*
>
> *— A. Michael Noll (1967)*



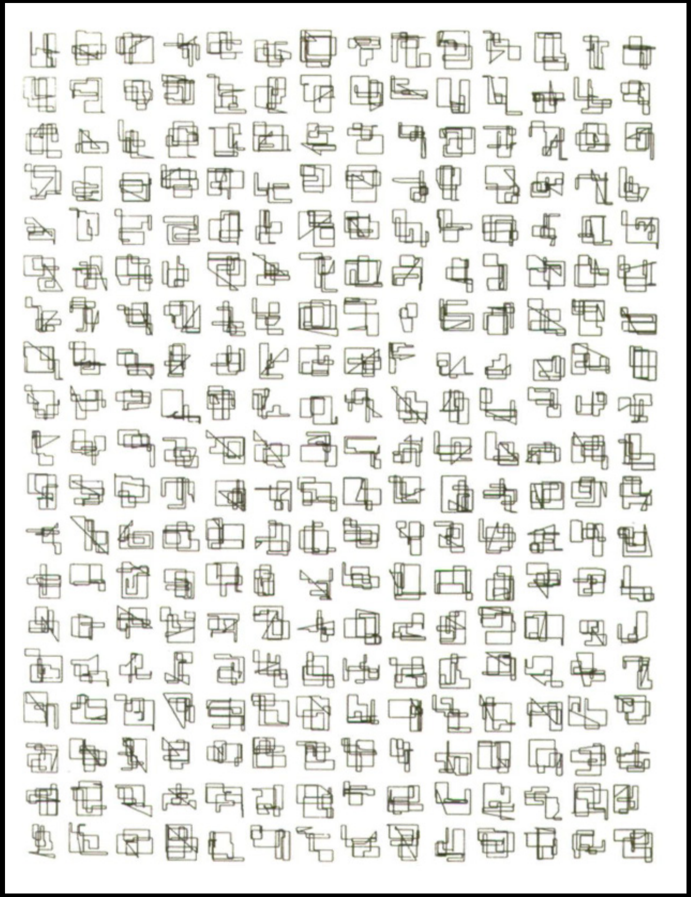
Gaussian-Quadratic (1963)
by A. Michael Noll

The name refers to the quadratic rule used to determine the start and end points of the lines.



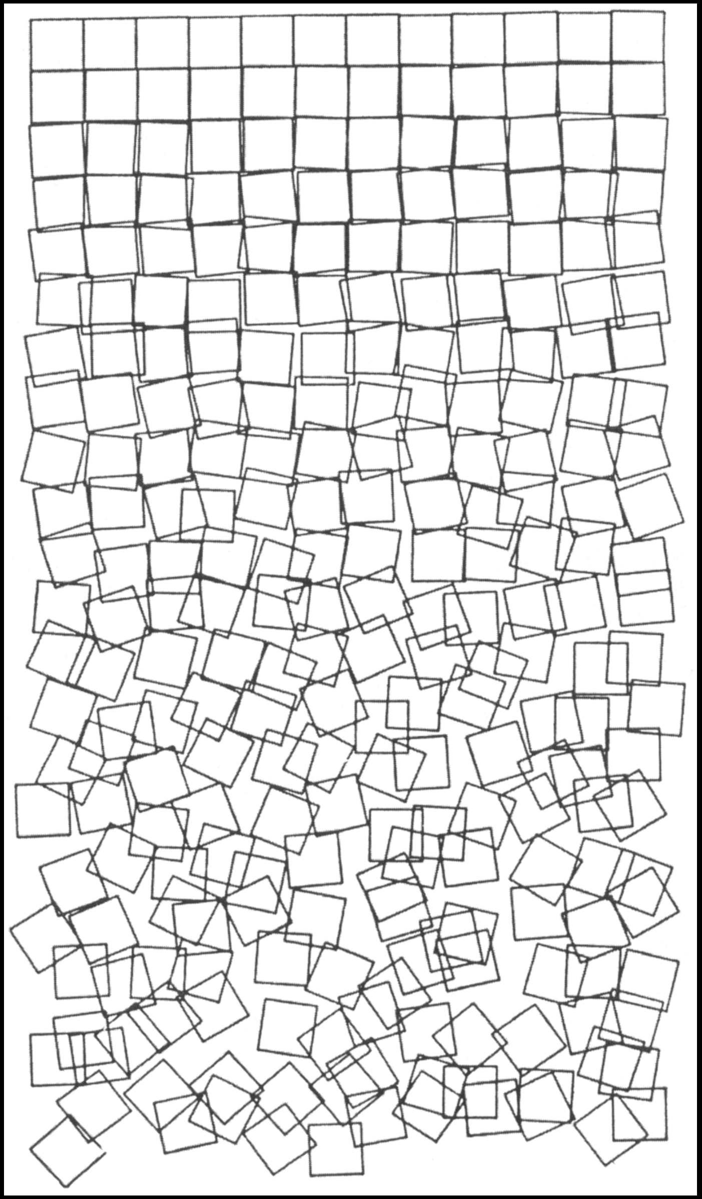
13/9/65 also called Homage à Paul Klee (1965) by Frieder Nake.

It has been described as incorporating some stylistic rules gained from analyzing drawings by Paul Klee, mainly in how it distinguishes micro- from macro-structures.



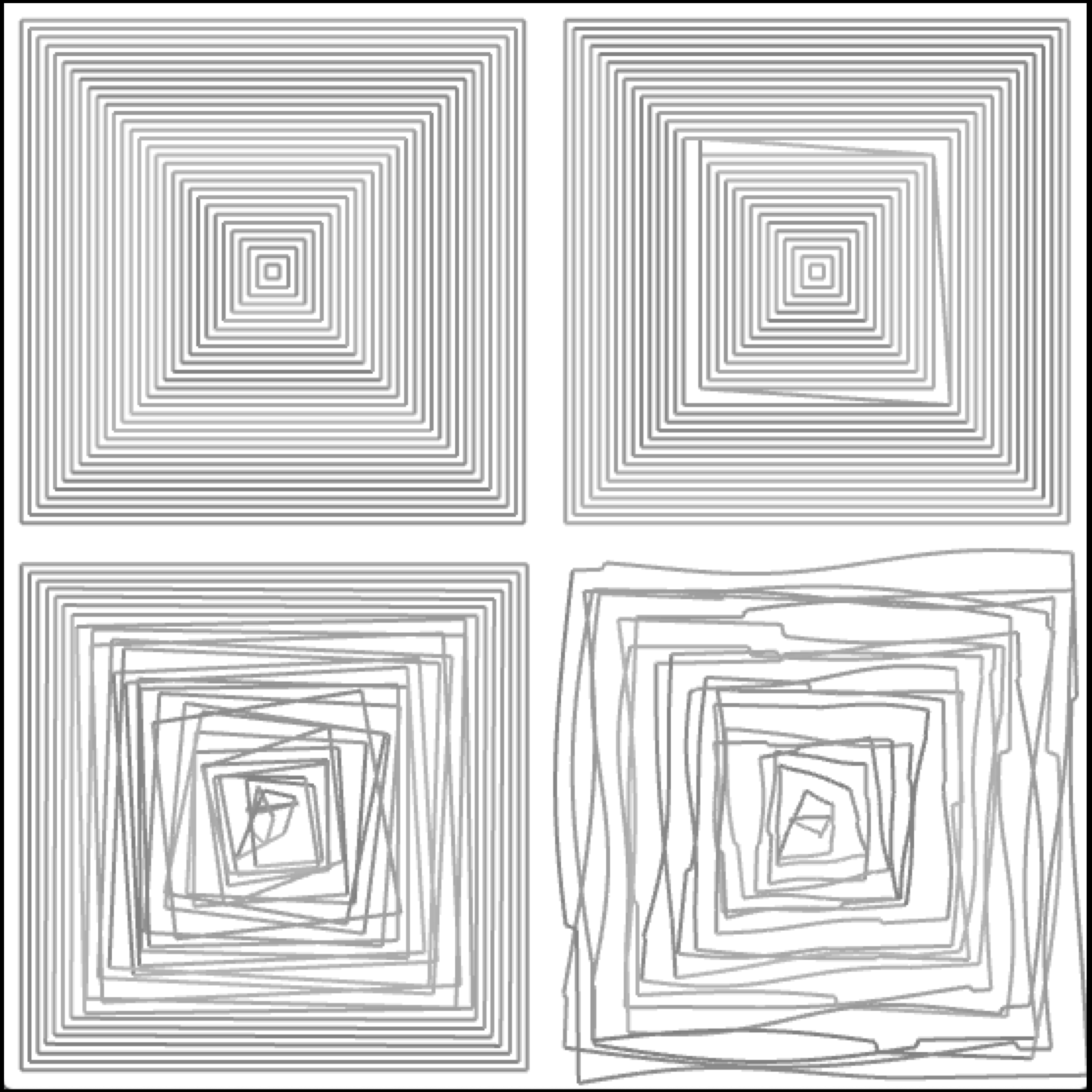
23-ecke (1965)
by Georg Nees

This series of pieces are all defined by the same rules — draw a line of arbitrary length and cardinal direction ten times, then connect the last line to the starting point.



Schotter (1968-1971)
by Georg Nees

As the program iterates through the rows, the coordinates of the squares become more randomized.



Computer Rosace-series (1974) by Vera Molnàr

This piece features four squares increasingly influenced by introduced noise, or controlled randomness.

“I use simple shapes because they allow me step by step control over how I create the image arrangement. Thus, I can try to identify the exact moment when the evidence of art becomes visible. In order to guarantee the systematic nature of this research, I use a computer.”

- Vera Molnàr

From Then to Now

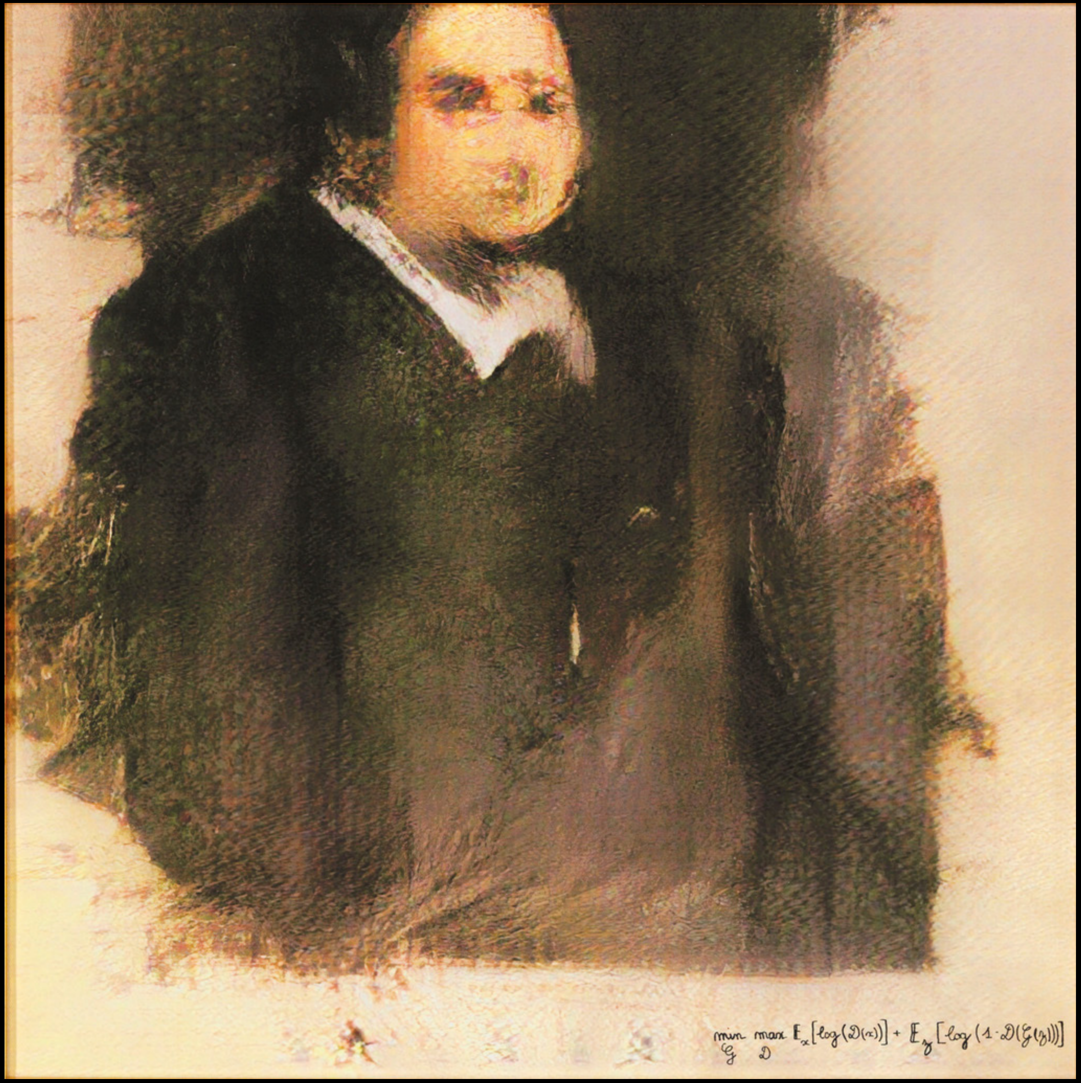
In the following decades, computer aided generation began to be employed for more practical purposes. Architectural firms sought algorithmic approaches to designing the layout of hospital floor plans. Decades later, in 2006, NASA used an algorithmic approach to examine millions of potential antenna designs before settling on a final one in a process patterned after Darwin's 'survival of the fittest'.

Generative design is a different way of thinking for designers. Instead of several one-off designs, one has to think in terms of a system for design.

Today, we have artificial intelligence to aid us in this process. The system is now able to evolve over the course of its lifetime, changing itself, and, with it, the output.

The Paradigm of Artificial Intelligence

When we speak of artificial intelligence we speak of machine learning, that is to say, algorithms that improve automatically through experience when trained on data. One of the main limitations that simply made machine learning unfeasible until very recently was processing speeds and the amount of data one needed to feed the algorithm. Now, we take the trove of information available to us on the internet for granted, but it has taken decades to build.


$\min_{G} \max_{D} \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1-D(G(z)))]$

Edmond de Belamy (2018)
by Obvious (collective)

Generated by a generative adversarial network (GAN), where two neural networks compete in a zero-sum game. The piece, being oil on canvas, sold for $432,500 at Christie's Auction House, manyfold its original estimate of $7,000-$10,000.

The Next Rembrandt (2016)
by Ing

An algorithmically generated portrait painted in the style of the Rembrandt, based on 346 paintings.

Not Just Rivaling; Besting

When it comes to the game of go, determining a winner is a simple matter; the game ends when both players pass two turns in a row, one player forfeits, or the board is full of stones. The number of stones in each players posession are then tallied and the player with the most wins. In 2016, the reigning world champion Lee Sedol was beaten by AlphaGo in a five-game match, and along the way AlphaGo played a surprising move that many of the professional analyzers at the time considered a mistake until it later became clear it was crucial to AlphaGo's victory. AlphaGo had deviated from the so-called 'local maxima' to reach new and previously unconceived heights. To many, it was considered a truly creative move and has since been widely adopted by human players. Neither Hassabis, the founder of DeepMind, the company behind AlphaGo, or the human sat at the tournament locale placing the stones for the algorithm, had conceived of the move; the algorithm had.

Identifying and deviating from local maxima in aesthetics is more difficult due to the subjective nature of determining worth. Picking out 'winners' is no longer as simple as tallying the stones on a wooden board. It is widely accepted Rembrandt was one of the most talented painters to have every lived, but defining *why* is no trivial matter and usually within the purview of art experts.

The Next Rembrandt sought to faithfully ape Rembrandts style. However, speaking from personal opinion, I am more likely to attribute the craftsmanship to the team behind the algorithm than the algorithm itself in this instance.

Mainstream Attention

AI generated artwork has become somewhat of a fad in the recent decade. First with Google's DeepDream which uses so-called convulutional neural network generation to search and find patterns where there are none, a process called pareidolia. The result is reminiscent of psychedelic art.

Similarly, wombo.art is another algorithm by the company wombo that scoures the internet for images matching the prompts entered. For instance, the prompt 'strawberry' feeds the algorithm the images of strawberries it finds on an engine search, which it is then trained to interpret. The result becomes an abstract amalgamation of the prompts entered.

DeepDream has been described as 'future kitsch'. While a neat display of the advancement of image recognition, the value of the pieces it produces are questionable — especially since everyone and their grandma would be able to generate a new piece after a quick visit to their website. But is there a usecase for such art? Perhaps DALL-E, one of the newest innovations within the field, could topple the doubters.

Deep Dream Vision of Van Gogh's Starry Night (2015) by Deep Dream

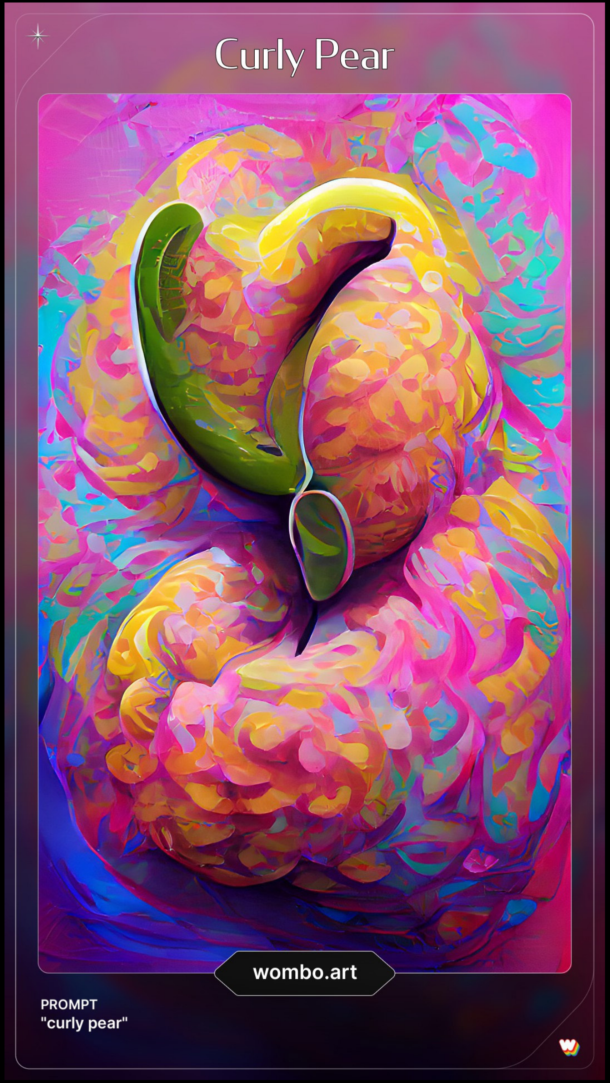
Google's Deep Dream algorithm interprets Starry Night. Posted by a user on reddit.com

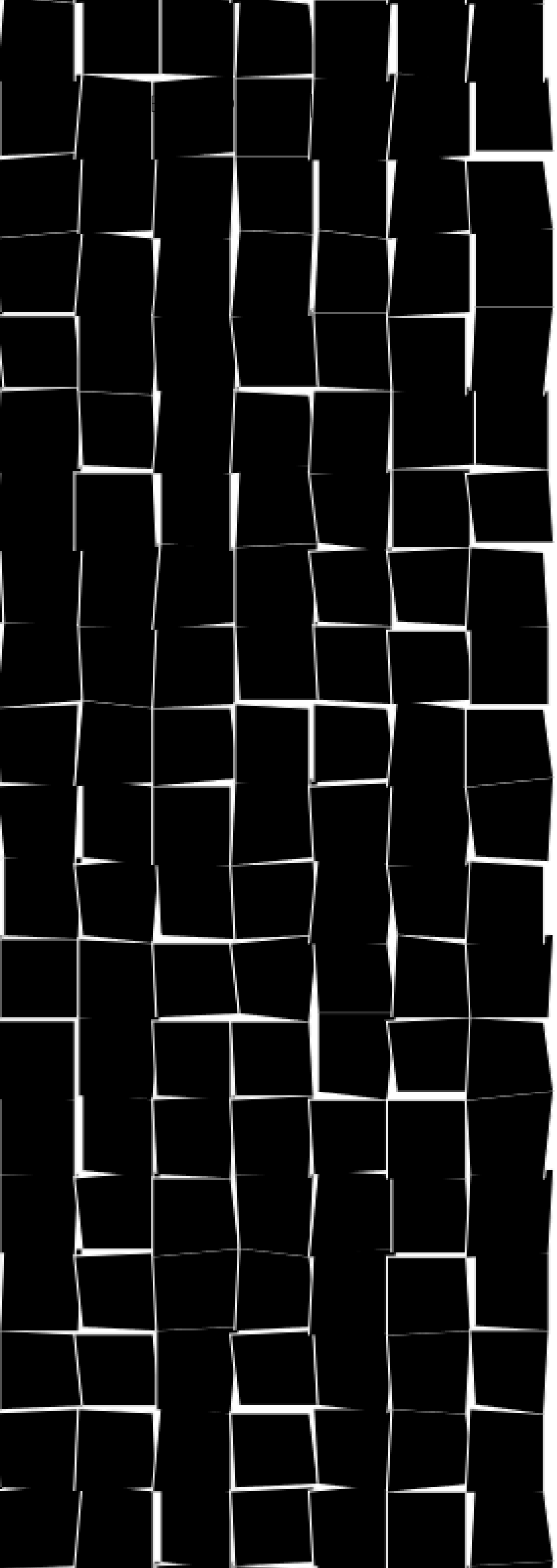
Avocado Chair (unknown) by DALL-E

An exploration of DALL-E's, OpenAI's image generator, ability to generate visuals from two unrelated concepts.

Curly Pear (2022) by wombo.art

Newly generated abstract artwork by the wombo.art algorithm based on the prompt "curly pear".





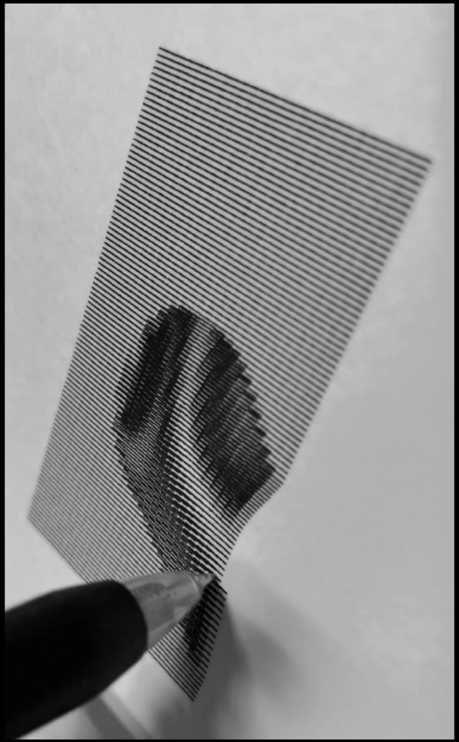
Contemporary Creative Coders

Not all algorithms are made for public usage, nor should they be; the parameters determining the output might not lend itself to enough interesting variance.

'Creative coding', as it is often referred to, is a practice familiar to many artists, and, to some, their predominant way to create works of art or design. The technique lends itself well to iterating through many variants that diverge either through randomness or through intentional changes to the parameters or logic. In short, it can be a way to create a large body of work very quickly.

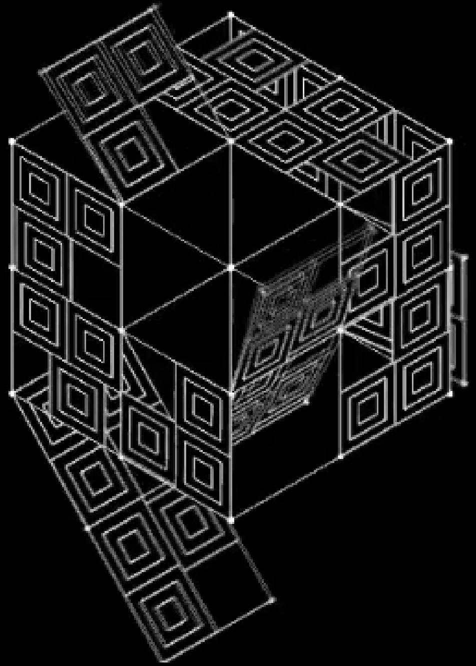
There exists many practitioners across the globe, all with their own unique expression. Pieces start simple and complexity arises after iterations and expansions.

Personally, I am more drawn to the geometric than natural shapes. which is why I find the pieces by Hyojung, Jacob, and Godil, among others, particularly inspiring. For this project, I will attempt to pursue something similar to both this and to that exhibited by the pioneers in the 60s. While I could show many examples vibrant in color, I intentionally chose to look at ones that only use two colors or are in black and white. And thankfully many of these creative coders have made their code available to the public to allow for insights into the underlying logic and often elegant mathematics.



Unnamed artwork (2022)
by Hyojung Seo

Hyojung Seo is a Seoul-based creative coder. This piece features topology through the use of varying line thickness and is a demonstration of Axidraw, a profesionally manufactured pen plotter.



Unnamed artwork (2022)
Etienne Jacob

Etienne Jacob is a french creative coder who focuses on geometric figures represented in 3D-space.

Unnamed artwork (2020)
Kjetil Golid

A 3D generation using the rendering software Blender.

04

Ideation

Starting Out

This project has been brewing in the background over the last year, starting with the independent project period last semester where I initially set out to explore creative coding as an artform. Since, I have sought to incorporate this more into the practicality associated with the field of design. I acknowledge, however, that I am not the best programmer and that many projects of this ilk would be time consuming and therefor outside the scope of this project.

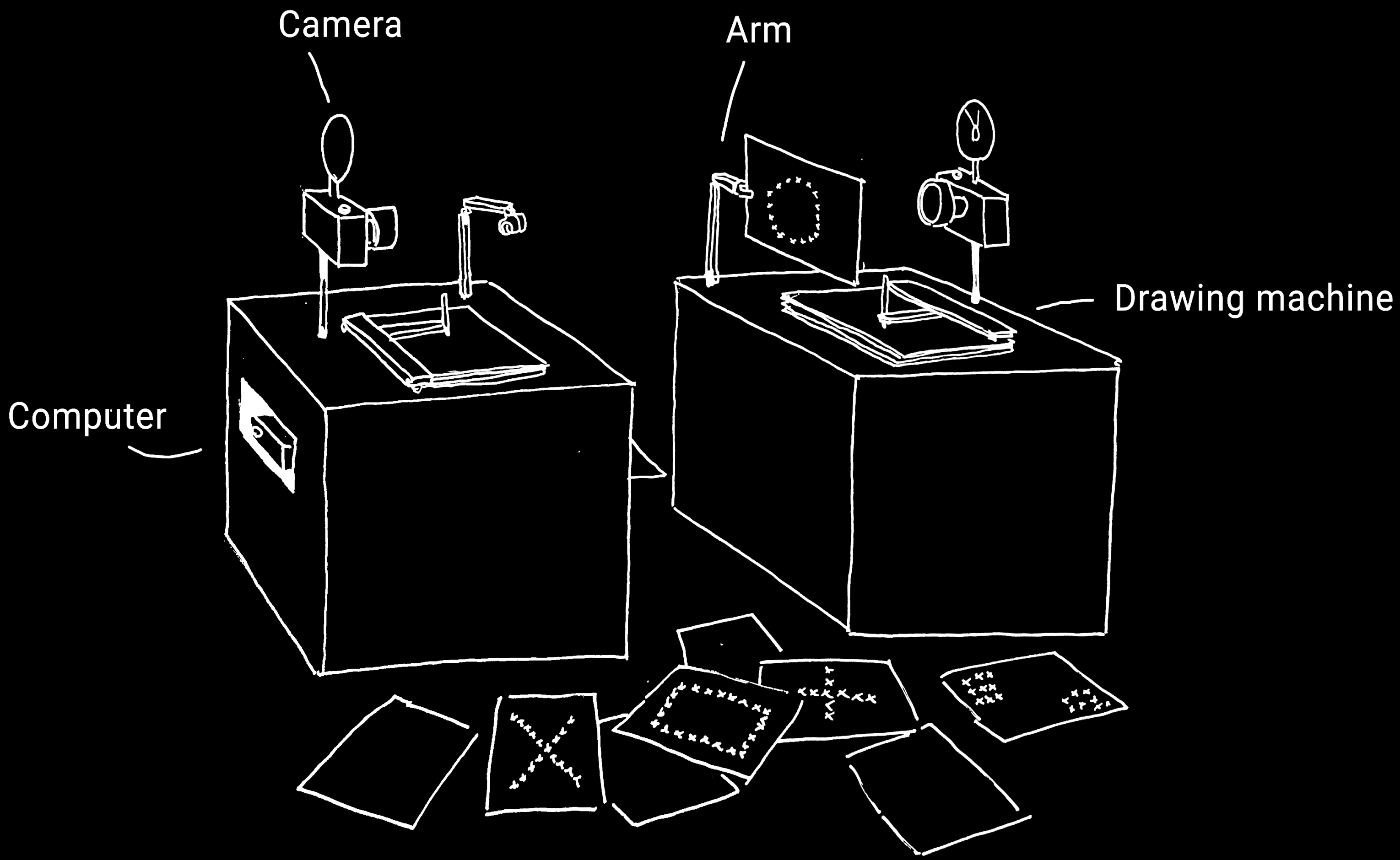
The Idea

Initially, I wished to construct two pen plotters which took turns drawing based on an algorithm and showing their drawing to the other pen plotter. The drawing would be captured by a camera, and a program would iterate over the pixels and generate a key based on which pixels were black and which were white. A key would then be generated to serve as the parameters for the algorithm, which would then instruct the pen plotter to draw something new. The process would then repeat ad infinitum. Or at least until the pen plotters ran out of paper or ink or were unplugged.

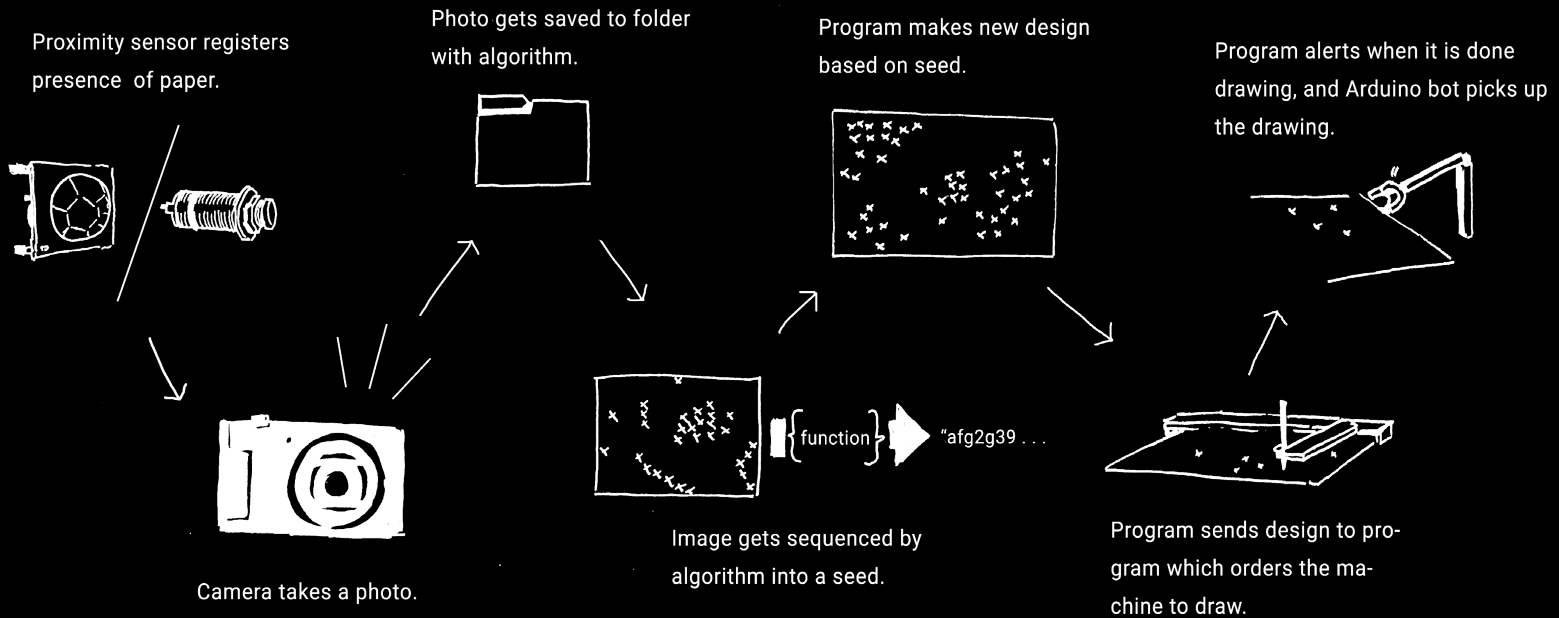
The core idea, when distilled, would be to humanize the algorithm responsible for the drawing and to bring into question what serves as 'inspiration'. Given a prompt, a human actor could come up with a variety of drawings, just like the pen plotter would. In reality, there is no reason for the pen plotters to exist in a physical space as showed in the sketch to the left — the same process could be displayed on a screen. Nor would there need to be two screens — two different programs could run in tandem on the same computer. And then, going even further, nor would there need to be two different programs — the same program could run two independant functions within the same file.

But for some reason, distilling it down to its minimum requirements would not have the same effect as witnessing two machines happily drawing off each other forever. By recognizing ourselves in them, we attribute them some of our own qualities.

What is perhaps even more interesting to consider would be exactly what separates two humans set upon the same task two machines?



A preliminary sketch of how I envisioned the system to operate. The arms are of course an incredibly far-fetched idea, even more so than the rest of the setup.



Proximity sensor registers presence of paper.

Camera takes a photo.

Photo gets saved to folder with algorithm.

Image gets sequenced by algorithm into a seed.

Program makes new design based on seed.

Program sends design to program which orders the machine to draw.

Program alerts when it is done drawing, and Arduino bot picks up the drawing.

The theorized loop. As with the hardware, there are many points of potential failure.

Points of Failure

This concept is wildly ambitious and completely unfit for the scope of this project. Not only would it require me to more than double the funds already invested, there are many moving parts, all of which must be operating smoothly for the process to run its course not just once, but continously. The mechanical components would require significant attention, as would the software. Even if javascript allows for the freedom necessary to automate the algorithm, feeding the pen plotter a simple drawing requires several interventions and is therefor far from an automized process.

It came as no surprise when I needed to downscale and brainstorm other approaches. The problem then became how I could maintain the core concept while shedding complexity.

Brainstorming

To come up with different ideas, I mapped my thoughts by hand. Metaphors arose along with different approaches and supplements to augment and clarify the original concept. The most prolific one, which ended up unused, was the analogy of the Chinese Room, a thought experiment posed by John Searle, in which a person assumes the role a CPU. Without any knowledge of chinese, the person is tasked with manipulating chinese symbols and numerals with the aid of an extensive manual. Once done, the person slips the appropriate string of chinese characters back underneath the door. Outside observers would mistakenly be led to assume that the person inside has knowledge of chinese.

While this tangents the core concept, the question of 'true understanding' could be a thesis in an of itself, and is certainly not within my ability to answer concretely. At best, the concept could be explored as a simile when expanding on other topics.

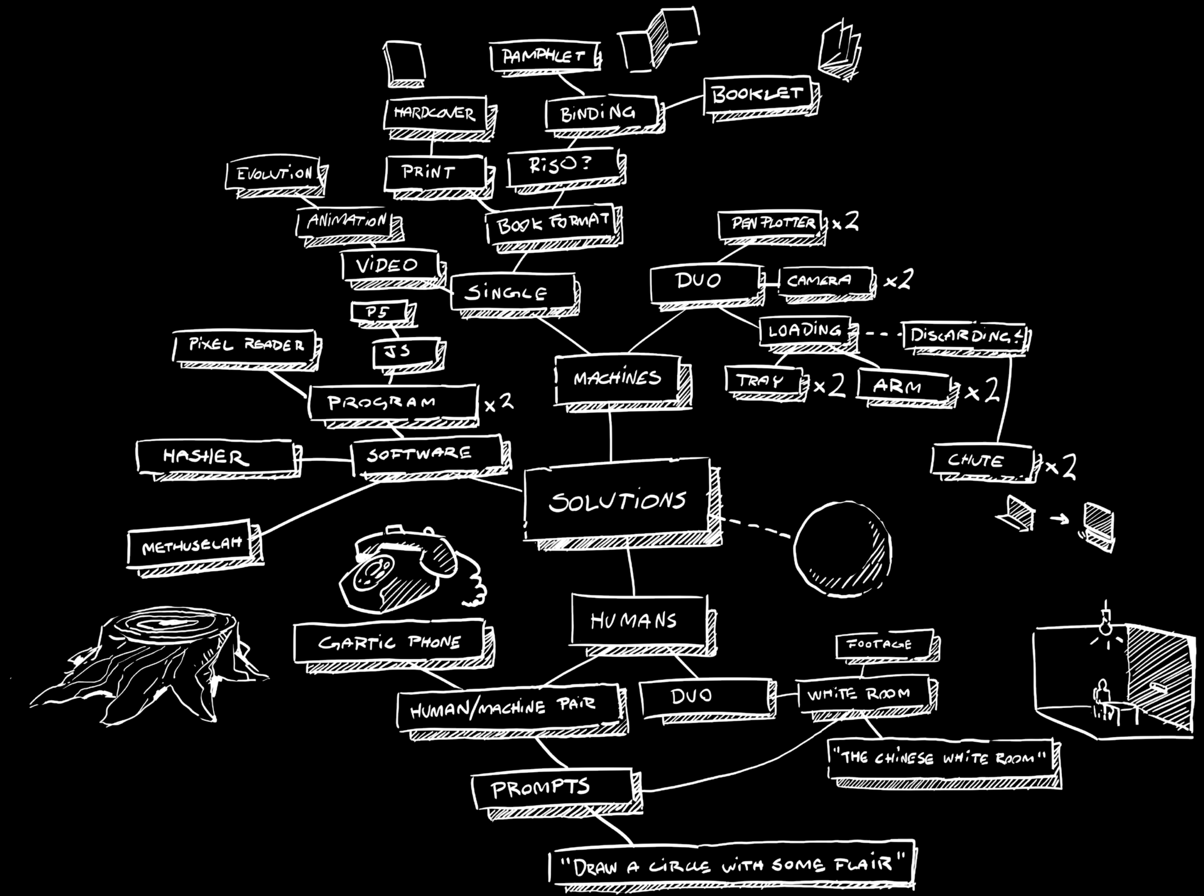
The two pen plotters drawing in tandem could be similar to the children's game chinese whispers / telephone, in which participants whisper a word to their neighbor, transmitting information in a chain often with interference occuring.

Formats

Is it the act of creation or the creation itself that is important? Or perhaps both? One could argue back and forth, but ultimately I am undecided. The question, however, is an important one since it will heavily influence the chosen medium for my exhibition and, similarly, the project as a whole.

To me, the evolution of the output is just as important as the act of creation. The need to exhibit a functional pen plotter would therefor be in contention. An alternative would be to display the output as a sequence, in either book format or any other. A video or animation could also be suitable.

However, I deem the observer's participation — the act of observing or intervening in the process — important, which means having a pen plotter performing the drawing live would be the best solution. It also helps communicate the idea better.



Brainstorming alternate approaches and related concepts.

05

Approach

Pen Plotter

The pen plotter was going to be integral to the project, so getting it up and running was my first priority.

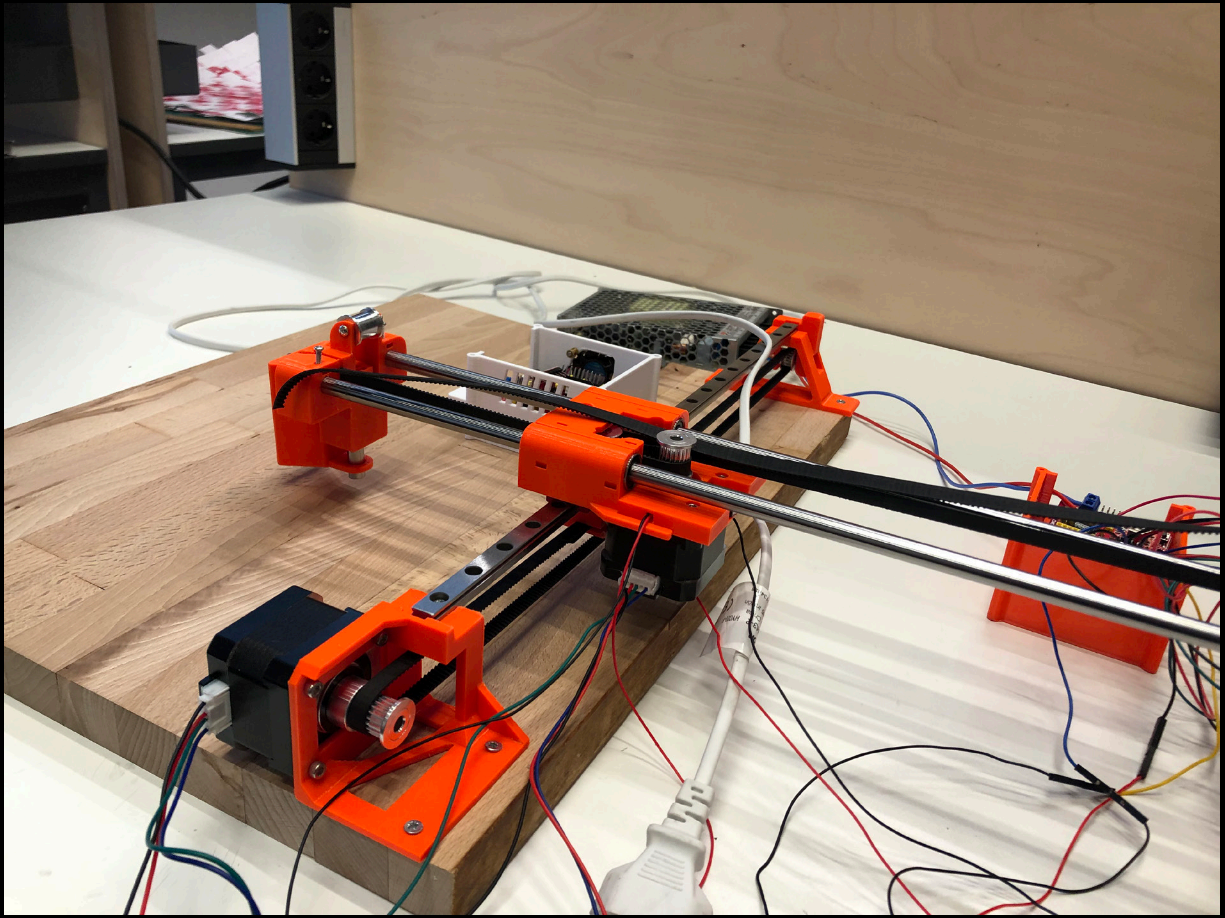
First off, I wish to mention that while the orange parts were 3D-printed by me, the designing of the parts was not done by me, but by Lewis of D.I.Y machines. Certain parts required alterations to fit the parts I had, which were either purchased or granted me by my institute.

Likewise, the assembly was made possible by tutorials filmed D.I.Y machines.

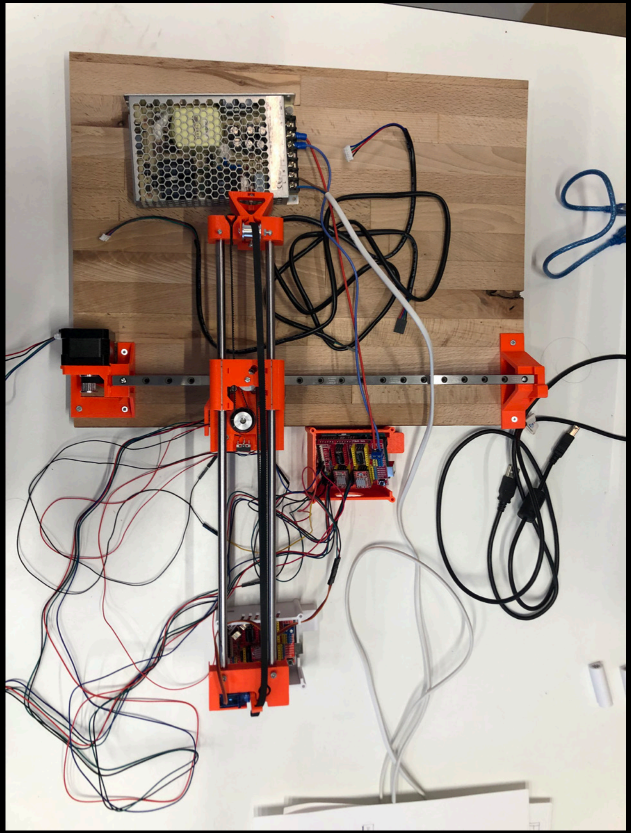
The construction of the pen plotter had started prior to the project period but was not finalized until late february due to various complications that took time to troubleshoot. Originally it would not work at all, but after identifying and fixing a problem with the wiring I got it running. Then it took some fine tuning to get to to behave satisfactory, and some further adjustments and amendments to the parts to attempt to hone its precision. While it still has some trouble, one major improvement was the exchange of the stepper drivers which powered the motors of the x- and y-axis for a higher end ones.

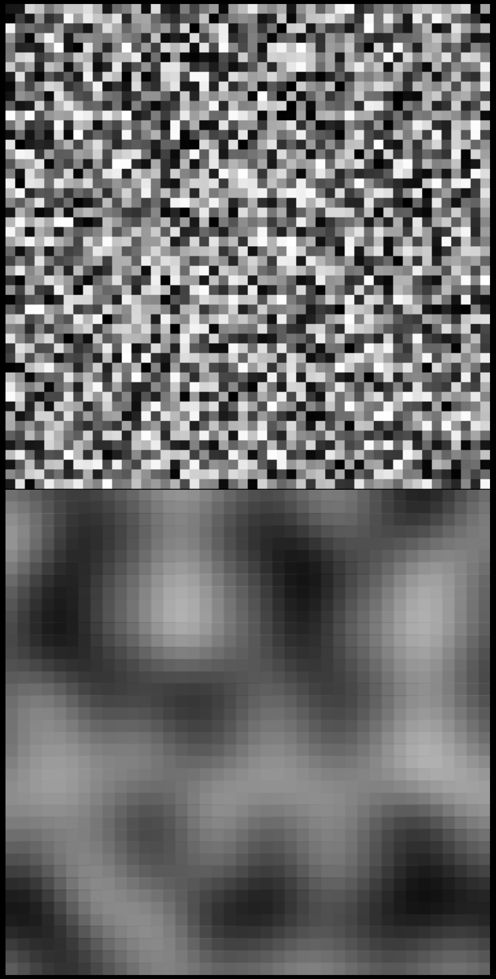
Here you can see a video of the pen plotter in action (https://www.youtube.com/watch?v=VW9CX-_-FZg).

The pen plotter is controlled by an Arduino board, which relies on Arduino software for commands. The program installed on it is a general purpose one that allows for the control of the stepper motors, with parameters like speed and increments per step (a step being one part of a full rotation, in our case 200 steps per rotation for the high-end motors). The more steps per rotation, the more granular the movement.



The Pen Plotter with all its parts. The power supply is large and clunky and due for an upgrade, but otherwise everything is finalized.





A visualization of Perlin Noise in two dimensions. The bottommost pattern relies on the Perlin algorithm to smoothen the randomness by interpolation. To generate static landscape, two dimensions are sufficient as each node in truth holds three values; the x-position, y-position, and intensity (represented here by a value between 0 and 255).

Expression

My work process mostly consists of sketching through code. That way, I can quickly generate many different iterations of a similar pattern either by changing the parameters or letting the randomness determine the output.

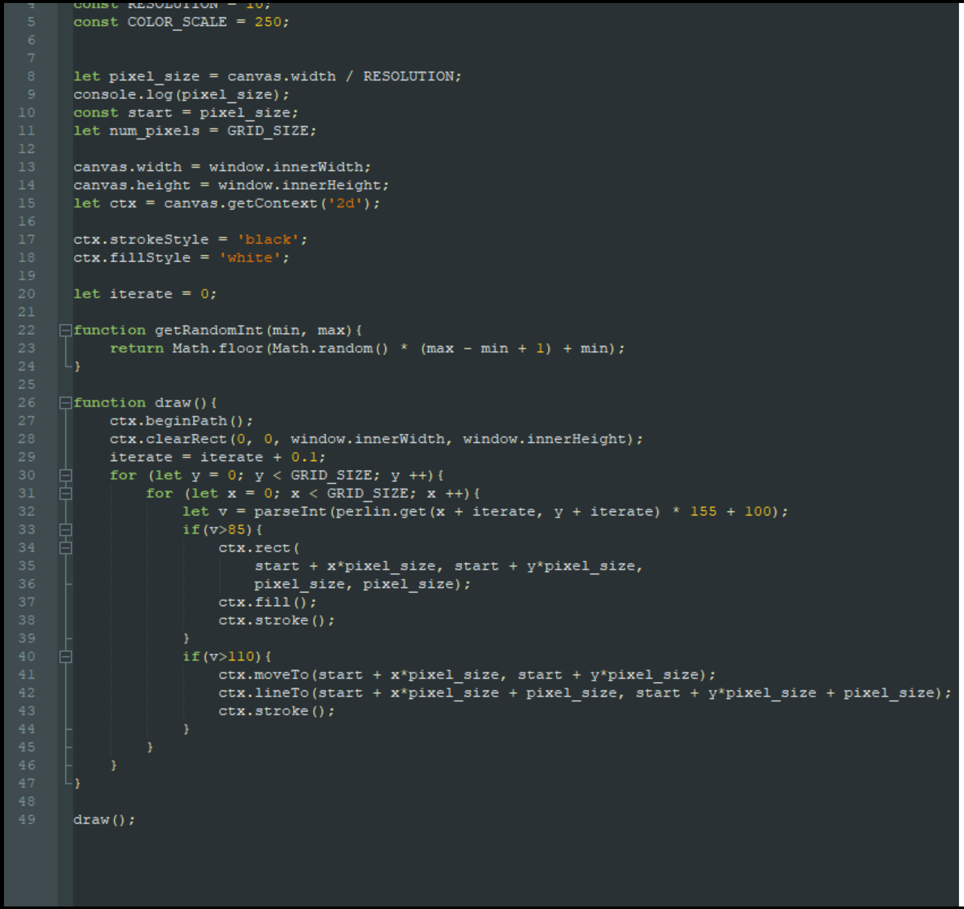
However, as stated by A. Michael Noll, pure randomness is often uninteresting.

Perlin Noise

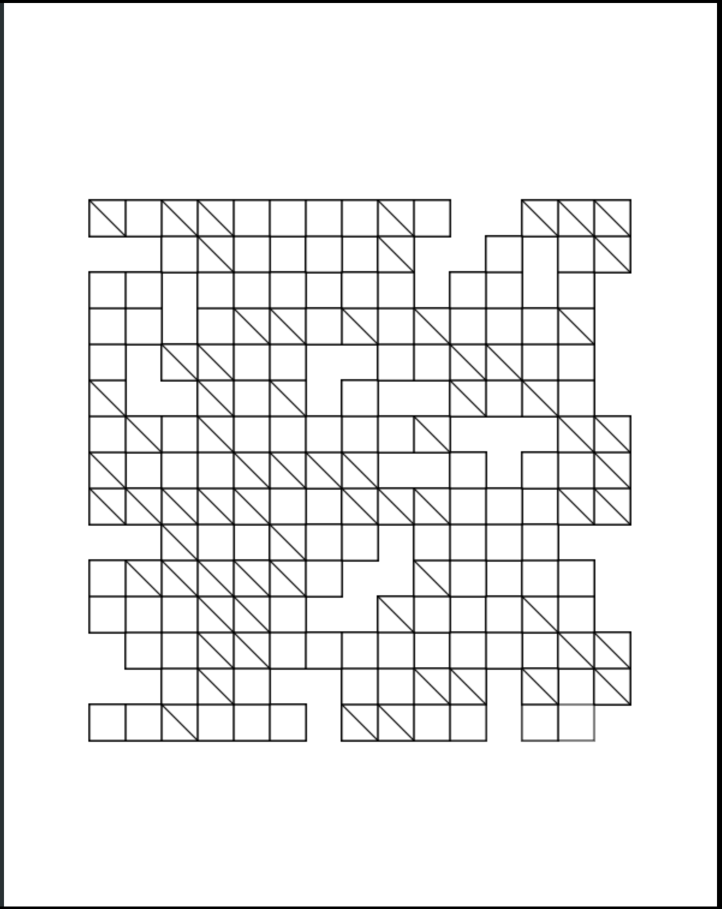
Perlin Noise is an algorithm that introduces local coherence between each point of randomness. Conceptualized and developed by Ken Perlin, it generates procedural texture, a type of gradient noise where a node's value is contingent on its neighbors. The noise becomes 'smoothed' out. It is often used to generate 3D environments akin to natural landscapes or 2D surfaces that appear organic.

Typically when I need randomness I use this algorithm instead of relying on the raw randomness procured by the computer. However, it does depend on the usecase; sometimes Perlin Noise is excessive.

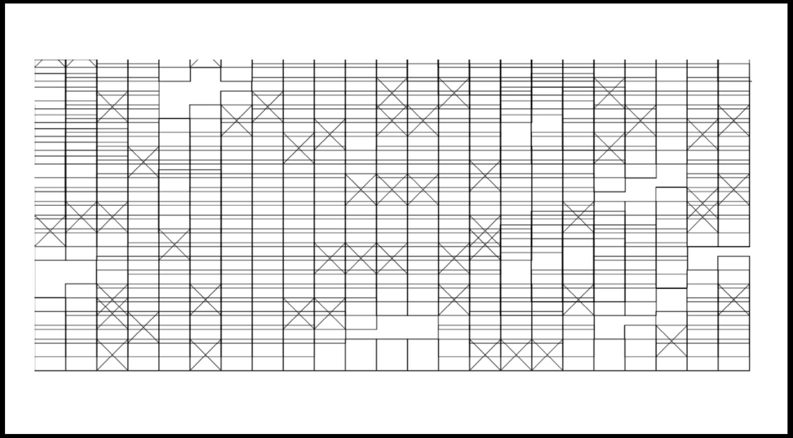
I rely on an externally implemented version of the algorithm. While I attempted to write my own, the result was slow, clunky, and scaled poorly when dimensions were added.



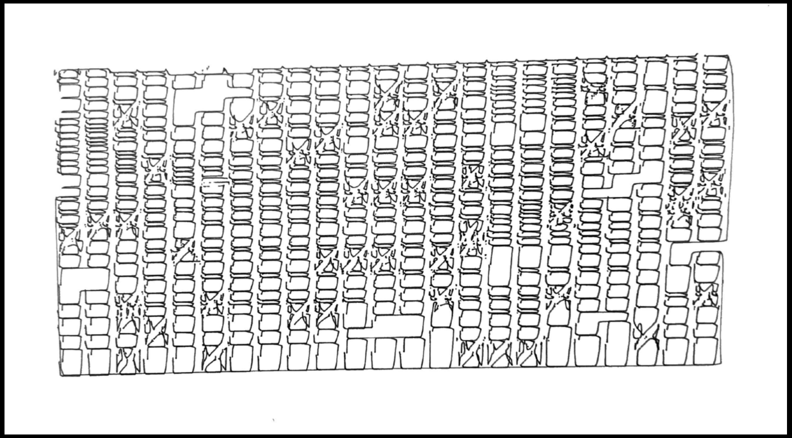
```javascript
const RESOLUTION = 10;
const COLOR_SCALE = 250;


let pixel_size = canvas.width / RESOLUTION;
console.log(pixel_size);
const start = pixel_size;
let num_pixels = GRID_SIZE;

canvas.width = window.innerWidth;
canvas.height = window.innerHeight;
let ctx = canvas.getContext('2d');

ctx.strokeStyle = 'black';
ctx.fillStyle = 'white';

let iterate = 0;

function getRandomInt(min, max){
    return Math.floor(Math.random() * (max - min + 1) + min);
}

function draw(){
    ctx.beginPath();
    ctx.clearRect(0, 0, window.innerWidth, window.innerHeight);
    iterate = iterate + 0.1;
    for (let y = 0; y < GRID_SIZE; y ++){
        for (let x = 0; x < GRID_SIZE; x ++){
            let v = parseInt(perlin.get(x + iterate, y + iterate) * 155 + 100);
            if(v>85){
                ctx.rect(
                    start + x*pixel_size, start + y*pixel_size,
                    pixel_size, pixel_size);
                ctx.fill();
                ctx.stroke();
            }
            if(v>110){
                ctx.moveTo(start + x*pixel_size, start + y*pixel_size);
                ctx.lineTo(start + x*pixel_size + pixel_size, start + y*pixel_size + pixel_size);
                ctx.stroke();
            }
        }
    }
}

draw();
```



Javascript (left) and the resulting output (right). This particular algorithm relies on Perlin Noise to determine where there will be gaps and which squares are granted strikethrough.



The output of a grid algorithm as displayed on the screen.



The same algorithm fed to and drawn by the pen plotter.

Geometric Simplicity

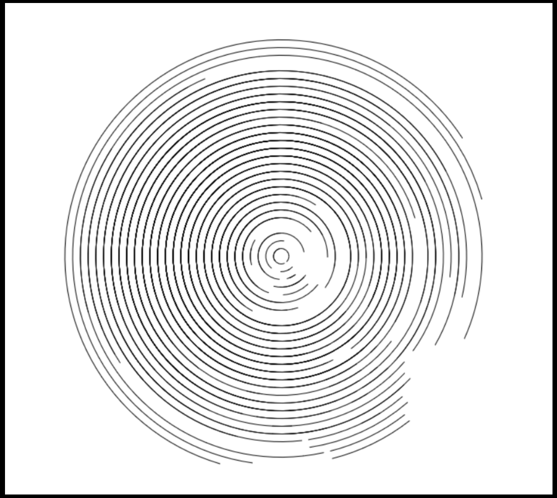
Personally, I enjoy the simplicity exhibited in work belonging to the early periods of computer graphic generation. Letting complexity arise from a few rules, as opposed to layering severeal rules on top of each, appeals to me. I appreciate the honety, the invitation of the observer to backwards engineer the underlying logic. It encourages speculation.

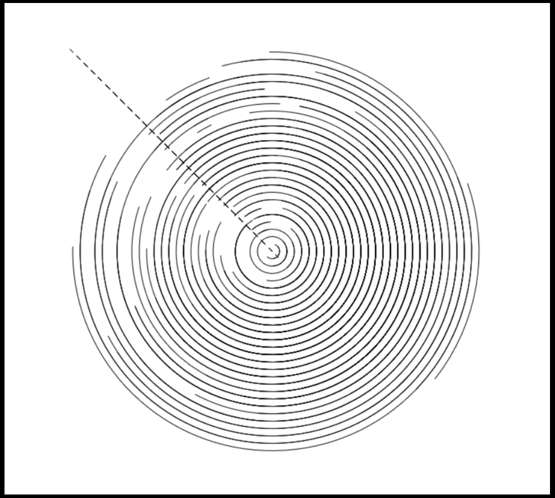
And there is no denying it; this approach is also suited to someone like me who is no genius when it comes to programming.

Perlin Noise runs contrary to this philosophy; I do not expect people to understand Perlin Noise as it is a rather complicated algorithm. I do, however, expect people to have an intuitive preference to Perlin Noise when compared to raw randomness.

Imperfection

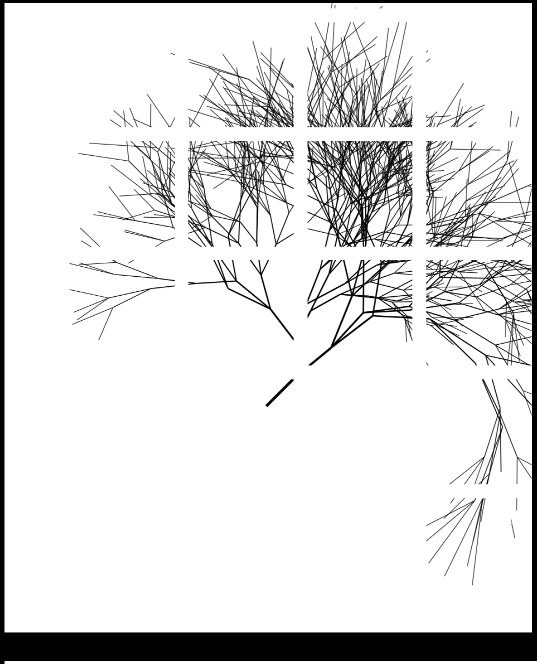
Along the way, the pen plotter revealed itself to be a faulty drawer. The part that housed the pen and allowed the movement in the z-axis (up and down; the disengagement and engagement of the pen) fit loosely in the part that in turn housed it. Therefor, *A radial experiment as displayed on the screen. The manipulation of squares and circles were my main focus of experimentation.*



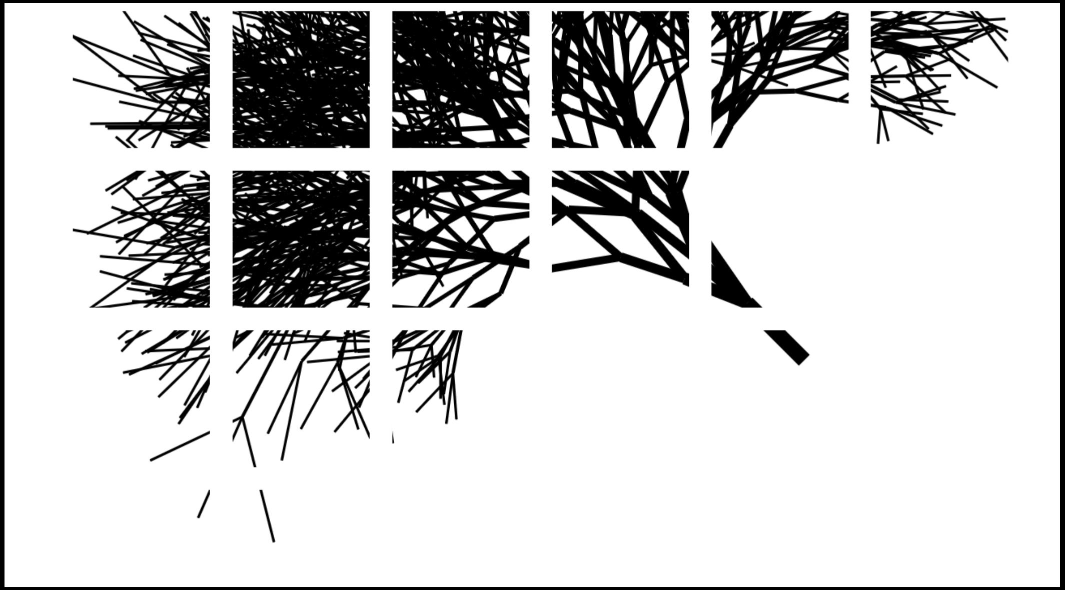


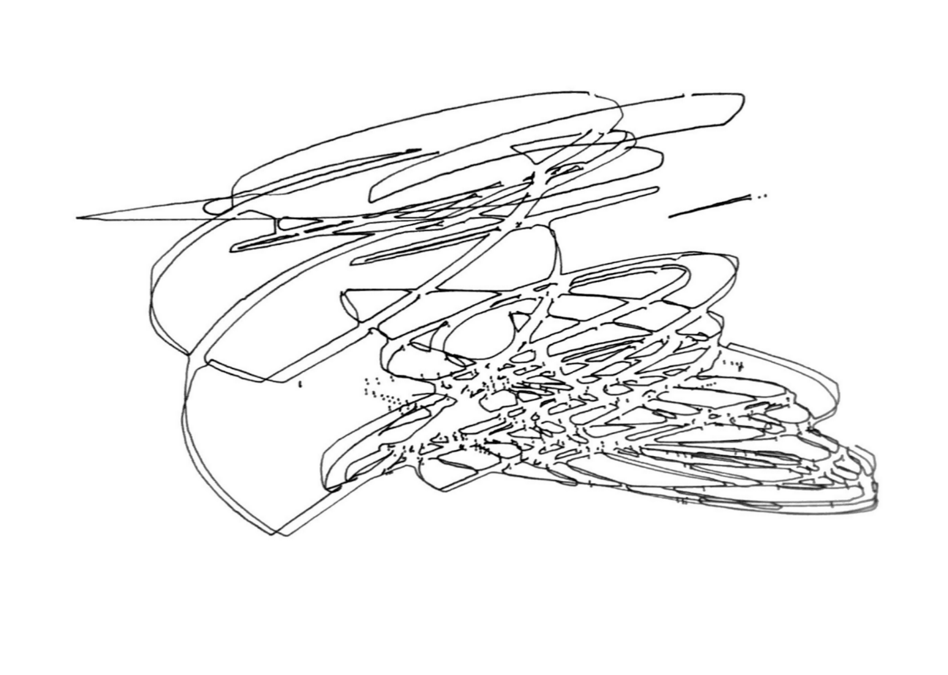
the pen was prone to wobbling. I intentionally left it like this as I was curious to how the expression would change due to this. At times it produced things I found interesting, at times it did the opposite. Fixing the inconsistency would prove far from trivial, however. It required drafting up a new part in Fusion, which I had little experience with, only to find out it would not solve the issue. Although, with some brute-force tinkering I managed to get it to behave better, albeit not perfectly.

I was at several points tempted to purchase an Axidraw despite its costliness.

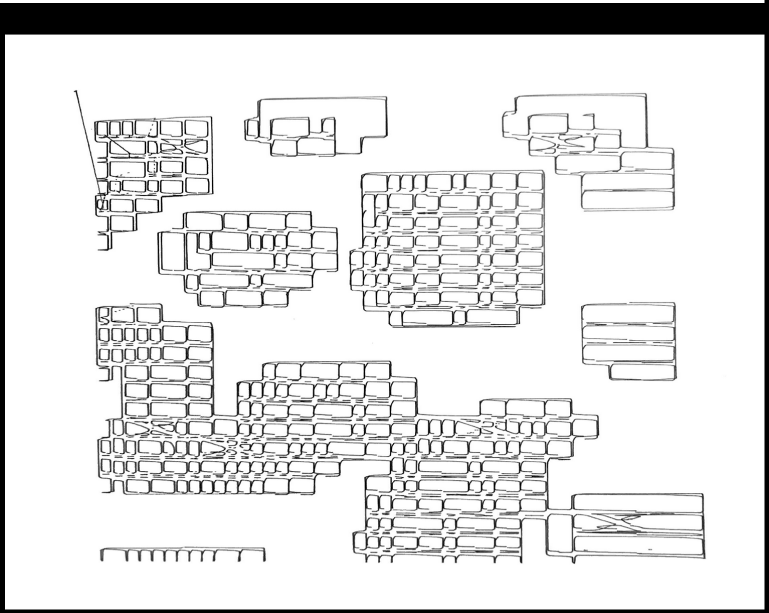


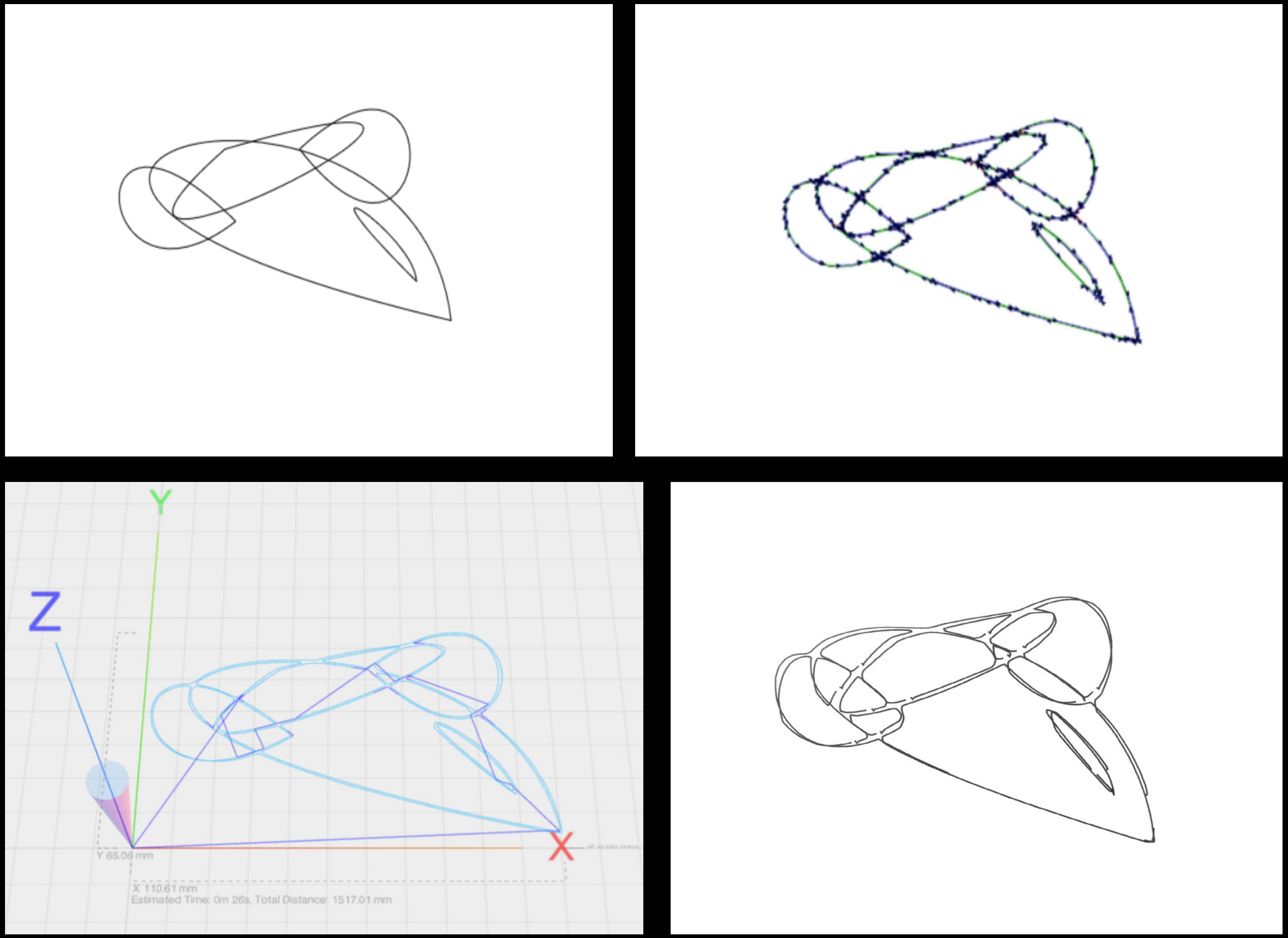
'Trees'. An experiment I was less satisfied with, but which posed a programming challenge.





Some more experiments with the pen plotter.





"Teardrops" as it moves through the various applications; first generated in the browser by javascript (top left), then transferred to Inkscape to be vectorized (top right), then uploaded to chilipeppr.grbl (bottom left), then fed to the pen plotter to draw (bottom right).

The Technical Pipeline

In order to have the pen plotter draw a picture, that picture must be fed through two different programs to prepare it. First it is generated, in this case through javascript but it could be through any method, digital or analogue. Then it is manipulated in a program called InkScape to translate the points on the drawing to a path, which can then be translated to a .gcode file through a third party extension. Then that .gcode is uploaded to chilipeppr, a program that provides a graphical user interface to allow for easier handling of the pen plotter. The alternative would be to issue it commands in the console of the Arduino, a process that would be tedious to put it mildly.

While the original output of the algorithm produced arcs of single lines, the obsolete versions of InkScape I had to use only allows for a path to be traced around every individual line, so you end up with a double set of lines in the .gcode. In this instance I was content with the result, as it is more visually interesting than what the algorithm produced.

To summarize, managing the pen plotter is like managing an unruly child; you can count on it to find new ways of acting up. From the feedback I received from both my peers, this unpredictability was of interest and worth pursuing. I, being the contrarian that I am, maintain that precision could prove just as interesting. Per the writing of this report it has been somewhat tightened up, though not entirely.

[redacted]

06

Discussion

Now, nearing the end, it would be beneficial to take a step back and evaluate my findings in regards to my initial research question; how can generative design challenge our perception of creativity?

Findings

As I initially theorized, relinquishing control in some ways can lead to sometimes surprising results. This was found intentionally in randomness and noise, and unintentionally in the form of the special expression the pen plotter decided to adopt. Due to the latter, I lost the ability to predict the overall quality of the outcome, something I am usually able to maintain when dealing with the varying outputs of an algorithm since the system is clearly defined. While I cannot predict the result of what will be displayed on screen down to every last pixel, I usually have some idea of what I am going to see.

It is important that I maintain the control of certain attributes, while the algorithm can take care of others. As I generate the visuals for this report, I am fully in control of how many squares will be in each row and column, but exactly how each square will diverge is up to the randomness of the algorithm.

It seems predictability is an important factor, which tracks with Boden’s definition of surprise.

I would argue most of what the pen plotter has produced has some novelty to it, but I will not comment too heavily on this as it is somewhat subjective. I will, however, part with a question; how novel is novel enough?

Evaluation the value of the output remains even harder. Seen by themselves, many of my experiments have little value aside from aesthetics, but seen within the context of the project another layer of philosophical value arises. To this, the pen plotter and the humanization of an artificial process contributes.

Shortcomings

Technical

My programming skills are unrefined, which influences my approach and also limits my capabilities. While I maintain drawing inspiration from the early works of the 60s as a suitable approach for a project of this scale, it is tempting to envision what a more complex approach may have looked like.

An Issue of Practicality

The issue of artificial creativity is far from a resolved one, and it would be hubris to claim to have the answer to such a complicated question. With that said, this thesis serves to raise questions instead of answering them, and one can argue the usefulness of such an approach.

Introspection vs. Outrospection

The project has been mostly introspective. While I theorized some approaches to include other human beings, such as through interviews or experiments where participants were to reenact the back-and-forth drawing the pen plotter(s) would be subjected to, I ultimately found it too unimportant to pursue. If I would have gone through with this experiment, it would have taken the form of two humans, which would then be juxtaposed the two machines, as opposed to a human and a machine cooperating. While I admit the latter could prove interesting, I personally do not find it to be on topic. It is the dynamic between the creator and the creation, and the blurring of those roles that is of importance. What level of autonomy must be reached for the creation to be attributed agency?

Broader Philosophical Quandaries

These types of discussions tend to gravitate toward the issue of sentience and consciousness. I have intentionally avoided them since they are too difficult to answer and prone to much pseudo-intellectual speculation.

The Way Forward

I would have loved to have more time to potentially realize my original vision, though I think this is unlikely to happen. Either way, I foresee I will continue to explore creative coding in my immediate future and on my own leisure.

Exhibition

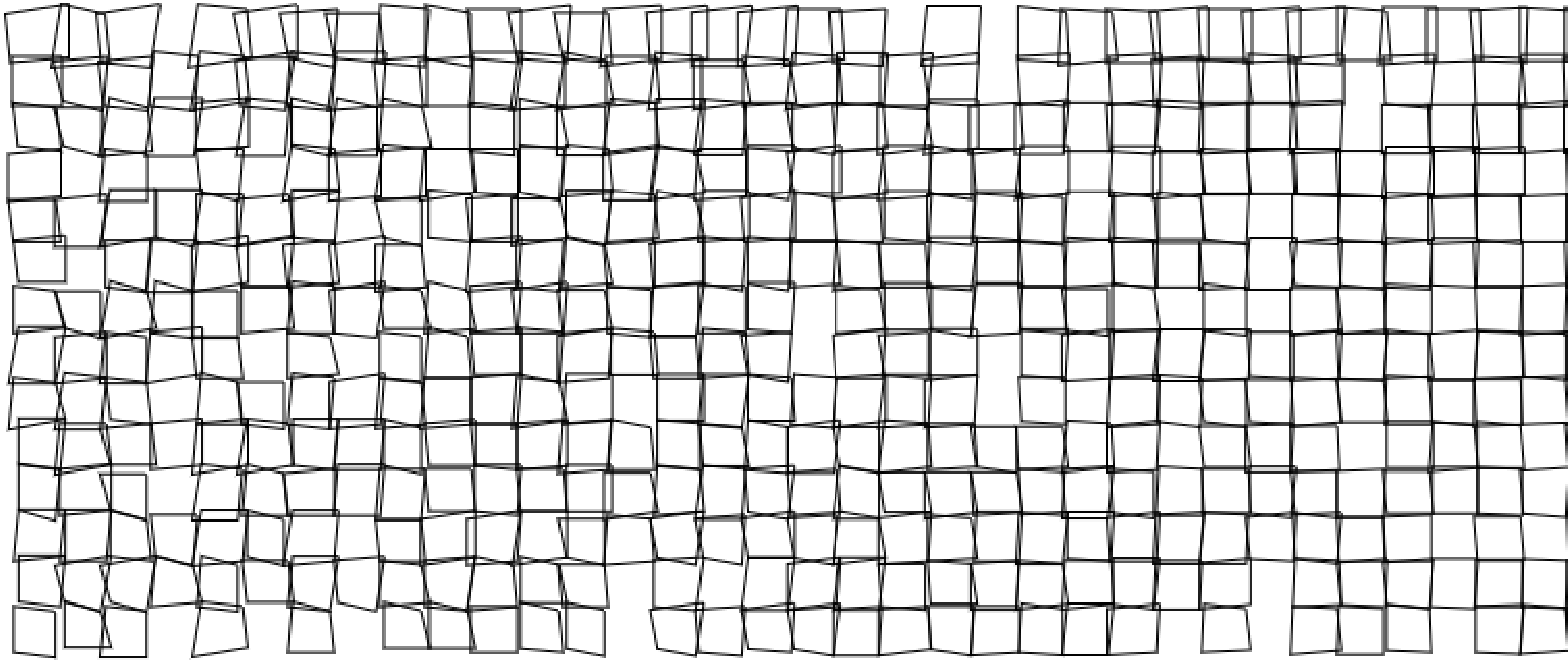
It remains important that the exhibition is fully autonomous, which means the pen plotter must be able to consistently draw independent of my interference. This, however, entails a number of challenges. How will the machine be fed paper, and how will it discard it? What will it do if its pen runs out of ink? How will it know when to start drawing and when to stop drawing?

The solution is to include the observer in the process. They will be responsible for providing the machine with paper from a stack I preemptively prepare beside it and they will order the machine to draw, perhaps in the form of a button press. This provides the added benefit of engaging the observer in the process. The machine will have to draw quickly, so as not to bore them, which means the drawings will have to be kept simple. It is also prone to abuse; a person could misunderstand instrocutions or intentionally fail to load the paper which would cause the drawing to scribble all over the the wooden board it is mounted upon.

And once it is done, it would be nice the participant could bring

However, this still appears like the best solution. Perhaps it could also be an experiment in human decency?

Nonetheless, it would be nice if the participants could take the drawing home with them.



07 References

Websites

Nake, Frieder. “Computer Art. A Personal Recollection - Researchgate.net.” researchgate.net. ResearchGate, June 25, 2014. https://www.researchgate.net/profile/Frieder-Nake-2/publication/221629559_Computer_art_a_personal_recollection/links/02e7e53aa798175bfd000000/Computer-art-a-personal-recollection.pdf.

Mcmanus, Ian Christopher. “Symmetry and Asymmetry in Aesthetics and the Arts - Researchgate.” researchgate. European Review, October 2005. https://www.researchgate.net/publication/4732061_Symmetry_and_asymmetry_in_aesthetics_and_the_arts.

Follett, Jonathan. “Ai, Architecture, and Generative Design.” Medium. Towards Data Science, November 13, 2020. https://towardsdatascience.com/ai-architecture-and-generative-design-e22320828d46.

Flynn, Meagan. “A 19-Year-Old Developed the Code for the AI Portrait That Sold for $432,000 at Christie’s.” The Washington Post. WP Company, October 26, 2018. https://www.washingtonpost.com/nation/2018/10/26/year-old-developed-code-ai-portrait-that-sold-christies/.

“AlphaGo.” DeepMind. Accessed May 2, 2022. https://www.deepmind.com/research/highlighted-research/alphago.

Nake, Frieder. “Construction and Intuition: Creativity in Early Computer Art.” researchgate. Computers and Creativity, August 2012. https://www.researchgate.net/publication/290813605_Construction_and_Intuition_Creativity_in_Early_Computer_Art.

Books

Boden, Margaret A. The Creative Mind: Myths and Mechanisms. 2nd ed. London: Routledge, 2005.

Denning, Peter J., and Matti Tedre. Computational Thinking. Cambridge, MA: The MIT Press, 2019.

Sautoy, Du Marcus. The Creativity Code: How Ai Is Learning to Write, Paint and Think. London: 4th Estate, 2020.

Images

Molnàr, Vera. Computer Rosace-Series. 1974. Dada.compart-Bremen.de. http://dada.compart-bremen.de/item/artwork/626.

Noll, Michael A. Gaussian-Quadratic. 1963. Dada.compart-Bremen.de. http://dada.compart-bremen.de/item/artwork/4.

Nees, Georg. Schotter (Gravel Stones). Researchgate. European Review, 1968. https://www.researchgate.net/figure/Schotter-Gravel-stones-Computer-generated-image-by-Georg-Nees-1968-1971-The-work-is_fig8_4732061.

Ropac, Thaddaeus. “Vera Molnár.” Thaddaeus Ropac. Thaddaeus Ropac. Accessed May 2, 2022. https://ropac.net/artists/231-vera-molnar/.

Edmond De Belamy. 2018. Obvious-Art.com. https://obvious-art.com/portfolio/edmond-de-belamy/.

Nake, Frieder. 23-Ecke. 1965. Researchgate. https://www.researchgate.net/figure/Georg-Nees-23-Ecke-1965-with-permission-of-the-artist_fig1_290813605.

The Next Rembrandt. 2016. Nextrembrandt. https://www.nextrembrandt.com/.

Deep Dream Vision of Van Gogh’s Starry Night. 2015. Researchgate. https://www.researchgate.net/figure/Deep-Dream-Vision-Of-Van-Goghs-Starry-Night-Reddit-2015-Accessed-March-24-2016_fig2_321557369.

Avocado Chair. Openai. Accessed May 2, 2022. https://openai.com/blog/dall-e/.

Curly Pear. May 1, 2022. Womboart. https://www.wombo.art/.
Hyojung, Seo. 2022. Instagram. https://www.instagram.com/p/Cck4n9qjETS/.

Golid, Kjetil. 2020. Instagram. https://www.instagram.com/p/CHnkPiNHhh6/.

Jacob, Etienne. n.d. Twitter. https://twitter.com/etiennejcb/status/1484633434777608200.