Generative || Scripted

Context

Generative en *Scripted* verwijzen naar twee verwante benaderingen in het creatieve proces. Vaak, maar niet noodzakelijk, denken we daarbij dan aan een context van kunst en technologie. Bij scripted kunst, maar misschien is de naam 'op instructies gebaseerde kunst' / 'Instruction based Art' wel juister, denken we aan kunst waarbij het creatieve proces wordt gestuurd door vooraf vastgelegde instructies, regels of scripts en scores. Dit kan gaan over traditionele kunstvormen zoals een muzikale opvoeringen, choreografieën, performances of beeldend werk, maar ook over digitale kunstwerken die door regels of algoritmes worden aangestuurd.

> *De twee pipes '||' in de titel staan voor de logische disjunctie of logische OR-operator. Deze geeft als uitkomst 'waar' (of true) als één of meer van de vergelijkingen waar is.*
> *De '&&' in vorige titel 'sound && music' is de logische AND-operator. Het geeft de uitkomst 'true' terug als beide vergelijkingen waar zijn, zoniet is het 'false'.*
>
> *Test: geeft de uitkomst van:*
> *a = 9;*
> *(a > 0 || a < 6);*
> *en*
> *(a > 0 && a < 6);*
>
> *&& en || noemen we boolean operators maar daarover later meer.*

Op instructies gebaseerde kunst, waarbij het geven van instructies zelf als artistieke handeling wordt gezien, vindt zijn oorsprong in Marcel Duchamps (1887 - 1968) radicale ideeën. Met zijn *readymade* sculptuur 'Fountain' stelde hij dat de keuze van een object al een creatieve daad was, waarmee hij de conventionele opvattingen over kunst uitdaagde. Deze verschuiving van kunstenaar-als-maker naar kunstenaar-als-kiezer of instructeur wordt beschouwd als het begin van conceptuele kunst.

Kunstenaars en componisten zoals John Cage (1912 - 1992), Steve Reich (1936), Merce Cunningham en Yoko Ono (1933) experimenteerden verder met toeval en instructie, wat leidde tot stromingen als Dada, Surrealisme, Fluxus en Minimalisme. Robert Rauschenberg en Sol LeWitt zijn bekende namen in deze context (weliswaar bekeken vanuit de dominante westerse kunstgeschiedenis).

> *Elke keer dat de muurtekeningen van Sol LeWitt worden getoond, worden deze direct op de muur getekend volgens een reeks instructies opgesteld door de kunstenaar. Deze instructies laten ruimte voor interpretatie, waardoor de uitvoerders zelf beslissingen moeten nemen over hoe lijnen en vormen geplaatst worden. Hierdoor ziet de muurtekening er telkens net iets anders uit.*

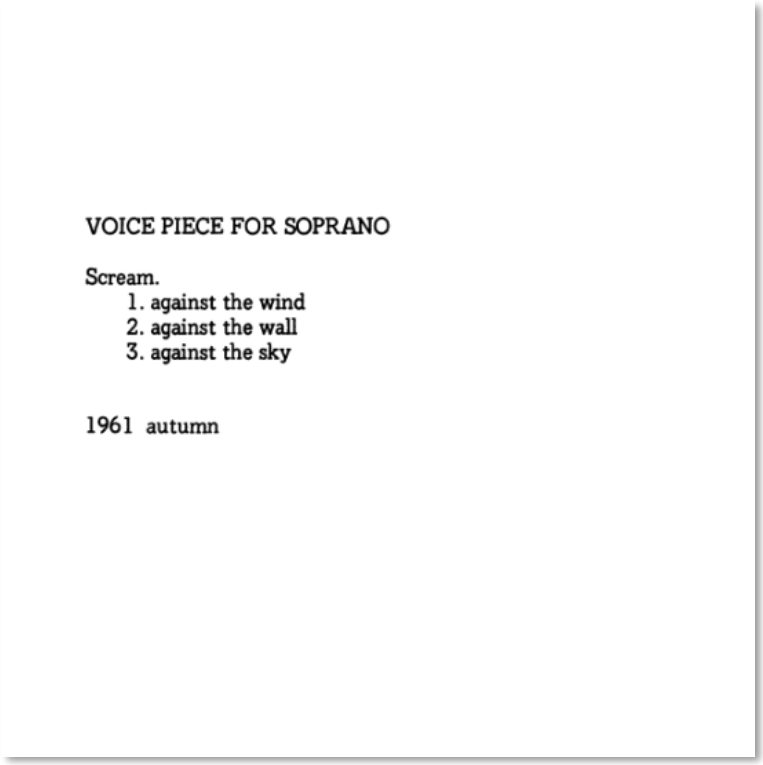
Wall Drawing 1180

Within a circle, draw 10,000 black straight lines and 10,000 black not straight lines. All lines are randomly spaced and equally distributed.

August 2005
Marker

Courtesy of the Estate of Sol LeWitt

Yoko Ono's instructies lezen vaak als gedichten (hoewel ze zegt dat ze dat niet zijn). De zinnen zijn duidelijk en beknopt, maar de instructies zelf kunnen variëren van surrealistisch en diepzinnig tot provocerend en alarmerend.



VOICE PIECE FOR SOPRANO

Scream.
1. against the wind
2. against the wall
3. against the sky

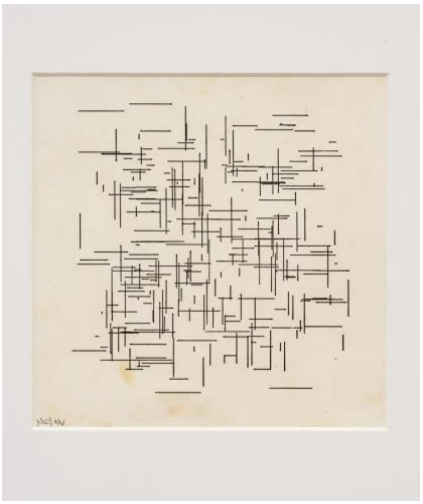
1961 autumn

Uit 'Grapefruit A Book of Instructions and Drawings' door Yoko Ono

In een moderne context, sinds de massale opkomst van computers en digitale technologieën, krijgt kunst en instructies nieuwe dimensies. Sinds midden vorige eeuw gingen kunstenaars voor het eerst computers gebruikten om visuele werken te genereren. Kunstenaars als Vera Molnár en Frieder Nake creëerden systemen of algoritmes die autonoom beelden produceerden. Hun kunstwerken werden gemaakt door een plotter die werd aangestuurd door een computer. De creativiteit lag in het schrijven van het programma, dat de acties van de plotter bepaalde. Deze stroming kreeg de naam 'Generative art' en 'Algorithmic Art' mee. Algorithmic artists werden Algorists genoemd.

Vera Molnár (1924 – 2023) is een van de eerste vrouwelijke kunstenaars die computers in haar werk gebruikte en wordt beschouwd als een pionier van de computerkunst. Al in de jaren 1950, voordat ze toegang had tot een computer, dacht Molnár als een computer. Ze ontwikkelde een systematische werkwijze, 'machine imaginaire' genoemd, waarbij ze stap voor stap regels volgde om haar geometrische tekeningen te maken. Deze benadering hervormde de traditionele visuele praktijken en legde de fundamenten voor computationele kunst. Molnár omarmde zowel de precisie als de snelheid van de computer, maar speelde ook met toeval en introduceerde *storingen* in haar algoritmes om voorspelbaarheid te doorbreken.

Vera Molnár



Molndrian, 74,066/13.36.22, 1974

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, ... which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.

Philip Galanter, 2003

Generatieve kunst ontstaat dus uit systemen of processen die vaak door algoritmes worden aangestuurd. De kunstenaar stelt hierbij bepaalde parameters of regels in, maar het uiteindelijke werk is dynamisch en kan verrassend of onvoorspelbaar zijn. Dit betekent dat generatieve kunst vaak ook interactief is en kan reageren op omgevingsfactoren, externe data of interacties met het publiek. Op deze manier wordt de dialoog tussen de kunstenaar en de technologie benadrukt. De kunstenaar creëert een kader maar het werk zelf komt door technologische processen tot stand.

Why is it so important for art to address contemporary materials? Because those materials have their own influence on our behaviors.

Uit 'The Importance of Generative Art' door Tyler Hobbs

Tyler Hobbs schrijft in zijn inleiding van bovengenoemd essay "Generatieve kunst kan aanvoelen als een technisch uitstapje met weinig relevantie voor ons dagelijks leven. Wat kan een computerprogramma te zeggen hebben over de menselijke ervaring? Zestig jaar geleden, in de kinderschoenen van algoritmische kunst, was er inderdaad weinig verband. Maar vandaag de dag kan het niet relevanter zijn. Het gaat nu rechtstreeks over een van de kernconstructies van ons digitale leven."

Ons leven wordt inderdaad steeds meer bepaald door de snelheid en verandering van media en technologie. We leven in een data gestuurde samenleving die voortdurend nieuwe vormen van interactie en sociale contexten creëert. In plaats van het verleden te idealiseren, is het misschien interessanter om ons artistiek domein aan te passen aan deze ontwikkelingen, de complexiteit ervan te omarmen, en zowel de schoonheid als de tekortkomingen te laten zien.

In the emerging, highly programmed landscape ahead, you will either create the software or you will be the software. It's really that simple: Program, or be programmed. Choose the former, and you gain access to the control panel of civilization. Choose the latter, and it could be the last real choice you get to make.

Stelt Douglas Rushkoff nogal bout in 'Program or Be Programmed'

Deze workshop is een praktijkgerichte benadering om het belang van programmeren te begrijpen. We gaan aan de slag met het lezen en schrijven van code. Het is essentieel om een beetje inzicht te krijgen in hoe software werkt omdat het geen neutrale tool is maar onze ervaringen en interacties dagdagelijks sterk beïnvloedt.

Maar we willen de opstap zo laag mogelijk en toepasbaar maken. Het is niet de bedoeling jullie af te schrikken. Om die reden hebben we onder andere voor de p5.js gekozen.

P5.JS

p5.js is een open-source JavaScript-bibliotheek die speciaal is ontworpen om *creative coding* toegankelijk te maken voor kunstenaars, ontwerpers, beginners, … iedereen.
Je kunt met p5.js bijvoorbeeld 2d en 3d animaties maken, werken met video en geluid, eenvoudige games of kunstinstallaties ontwikkelen.
Omdat p5.js in de webbrowser draait kan alles wat je maakt direct gedeeld en gepubliceerd worden op het web.

Maar p5.js is ook een project dat de nadruk legt op inclusiviteit en toegankelijkheid binnen de gebruikers- en ontwikkelaarsgemeenschap. Traditioneel zijn vrouwen en mensen van kleur ondervertegenwoordigd in zowel kunst als technologie, en pogingen om dit te verbeteren worden vaak als bijzaak gezien. Bij p5.js zijn diversiteit en inclusie kernwaarden, waarbij het project sterker wordt door de brede variatie aan ideeën en perspectieven van degenen die eraan meewerken.

Laten we starten met maken van een generatieve tekening zonder computer

Benodigdheden:

- papier
- een potlood
- drie kleurpotloden of stiften van verschillende kleuren

Regels:

1. Teken met het potlood ergens op het papier **een driehoek**.
2. Teken, nog steeds met het potlood, **een cirkel** in de overgebleven ruimte.
3. Teken als laatste **een vierkant** in de overgebleven ruimte.
4. **Kleur** één van de vormen in met een bepaalde kleur.
5. **Kleur** één van de twee overgebleven vormen in met een andere kleur.
6. **Kleur** nu de derde en laatste vorm in met een andere kleur.

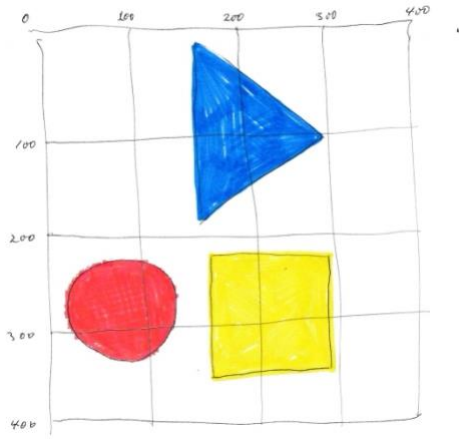
> 👉 Kijk zeker eens naar het project conditional design als je van regels, interactiviteit en spel in ontwerp houdt.
>
> 👉 ●■▲ door Bruno Mari

Nu gaan we onze tekening vertalen naar code met p5.js

Voor we de tekening omzetten naar code kunnen we best een paar hulplijnen plaatsen.

1. Teken met een potlood een vierkant om je drie vormen. Hoe kleiner het vierkant, hoe beter. Het moet alle vormen omsluiten.
2. Trek een lijn van het midden van de bovenkant van het vierkant naar het midden van de onderkant, waarbij je de tekening in twee deelt.
3. Trek een lijn van het midden van de linkerkant van het vierkant naar het midden van de rechterkant, en verdeel de tekening in vieren.
4. Trek lijnen die elk van de kwarten in vier vierkanten verdelen, zodat je een raster van 16 vierkanten krijgt.
5. Als laatste kan je afmetingen bij de lijnen plaatsen. Linksboven is punt 0. De uiterst rechtse (verticale) lijn is 400, de onderste lijn is ook 400. De tussenlijnen zijn op 100,200 en 300 verticaal en horizontaal.

Je schets moet er ongeveer zo uitzien:



👉 p5.js tutorial 👈

👉 P5js code collection 👈

P5.js editor user interface

1. Menu bar: gebruikt om schetsen op te slaan, te openen, te delen en te downloaden.
2. Control bar: bevat de knoppen Start & Stop om de schets te bedienen.
3. Edit pane: gebruikt om de p5.js-statements in te voeren die een sketch vormen.
4. Console pane: gebruikt om status- en systeemberichten weer te geven.
5. Preview pane: gebruikt om het visuele resultaat van de sketch weer te geven.

Maak een account aan

New sketch

Een programma in p5.js noemen we een schets.

setup() en draw()

Alle sketches zullen bestaan uit de functies setup() en draw():

- function setup() wordt één keer uitgevoerd als de *startknop* wordt ingedrukt.
- function draw() wordt keer op keer uitgevoerd aan 60fps (indien mogelijk of anders bepaald) (en tot de *stopknop* wordt ingedrukt).

Over het algemeen wordt een functienaam gevolgd door haakjes. Daarin kunnen vaak parameters meegegeven worden.
setup en draw bieden plaats aan meerdere functies. Deze worden gegroepeerd binnen accolades (curly braces).

createCanvas()

createCanvas(400,400);
Maakt een canvas van 400 bij 400 pixels.
Functies worden steeds afgesloten met een ‘;’ (semicolon)
Bij namen van functies bestaande uit 2 woorden, als noStroke() wordt het 2^{e} woord steeds met hoofdletter geschreven.

p5.js Online Reference

https://p5js.org/reference/

HTML, CSS & JS (javascript)

Een p5.js schets bestaat eigenlijk uit 3 documenten: een html pagina, stylesheet en javascript document.

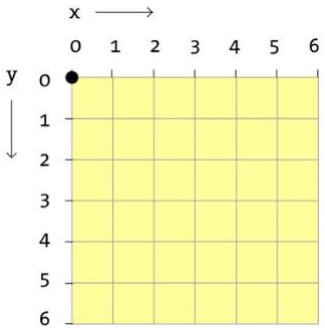
1 Triangle Square Circle First Steps

triangle() square() circle()

Zolang onze schets statisch blijft kunnen we alles toevoegen aan de setup functie.
De volgorde van functies is van belang.

coördinaten

Het canvas heeft een coördinatensysteem met in de linkerbovenhoek het punt 0,0 of 0 op de horizontale X as en 0 op verticale Y as.



2 TSC Canvas Color & Order

background(), fill(), stroke(), noStroke(), noFill(), strokeWeight()

kleuren

Kleuren kunnen op verschillende manier genoteerd worden.
Meest voor de hand liggend (R,G,B) met waardes van 0 tot 255
En (R,G,B,A) waarbij A staat voor alpha staat voor transparantie of dekking . 0 is volledig transparant, 255 is volledig dekkend.
Google Color Picker

Andere figuren: point, line, rect, ellipse, arc, vertex

3 TSC Random Variable Print

random()

int vs float

variables

moeten gedeclareerd worden in een bepaald formaat, bijv. let x = 0

- het sleutelwoord let, dit komt letterlijk van *let x be 0*

- de naam van de variabele (deze mag jij kiezen!)
- een isgelijkteken (=)
- een waarde die je wilt toekennen aan de variabele

global vs local

int vs floating point numbers
int() en floor() om een decimaal getal af te ronden.

print()

🎨 4 TSC Random Framerate noLoop

tijd voor beweging

we verhuizen onze code / functies naar de draw-loop

frameRate()

noLoop()

🎨 5 TSC mouse rectMode width/height

Interactie met de muis

mouseX & mouseY (systemvariables)

width & height

de breedte en hoogte van het canvas in pixels (ook systemvariables)

rectMode(CENTER)

function mousePressed()

is wat men noemt een eventhandler. Het wordt aangeroepen elke keer de user de muisknop indrukt. Zie ook keyPressed()

🎨 6 TSC Linear motion & Conditional Statements

lineaire beweging

conditional statements

of the IF Statement.
conditional statements of voorwaardelijke verklaringen bepalen wanneer specifieke regels code worden uitgevoerd. Deze test zorgt ervoor dat onze vormen binnen het canvas blijven.

🎨 optioneel 7 TSC bouncing FS

🎨 8 TSC circular motion

sin() & cos()

map()

herschaalt data

9 TSC a simple while loop

10 TSC a simple for loop

11 TSC nested for loops

nested for loops

12 TSC transformation

translate()

rotate()

angleMode(DEGREES)

push() & pop()

bonus:
export as png, svg & gif
interface sliders & buttons
simple dom elements

Verder leren & lezen

Meer links mbt generatieve kunst & *creative coding*

Video's en lessen

The Coding Train P5js track met Daniel Shiffman
The Anatomy of a p5js Sketch - A tutorial for new coders van Paul Wheeler
Six Features of Programming Languages - demonstrated in p5.js van n1ckfg
p5.js demos by Matt DesLauriers

Boeken

Aesthetic Programming: A Handbook of Software Studies door Winnie Soon & Geoff Cox
Generative Design door Benedikt Groß
Getting Started with p5.js door Lauren McCarthy
Computational Drawing Book door Carl Lostritto

Communities

Open Processing is een online platform en community gericht op het delen en creëren van digitale kunst, interactieve schetsen en creatieve codeerprojecten

Generative & Computer Art History

The ReCode Project, an active archive of computer art is a community-driven effort to preserve computer art by translating it into a modern programming language.
Generative Art Timeline Ontdek de 70.000-jarige geschiedenis van generatieve kunst door middel van honderden mijlpalen en tien hoofdstukken (door Peter Bauman)
Computer Grrrls Women & machines timeline door Marie Lechner & Inke Arns
A Longer History of Generative Art Nick Lambert, een van 's werelds meest vooraanstaande wetenschappers op het gebied van computerkunst, schetst de geschiedenis van generatieve esthetiek tot aan NFT's.

Generative Art blogs

Generative Artistry Tutorials & Podcast
Tyler Hobbs
Matt DesLauriers

Math

Sine / Cosine Reference
Sine and Cosine Calculator
Linear Interpolation — Introduction to lerp

Verwante kunstboeken en zines

Circle, Square, Triangle by Bruno Munari
The ABCs of Triangle, Square, Circle: The Bauhaus and Design Theory by Ellen Lupton