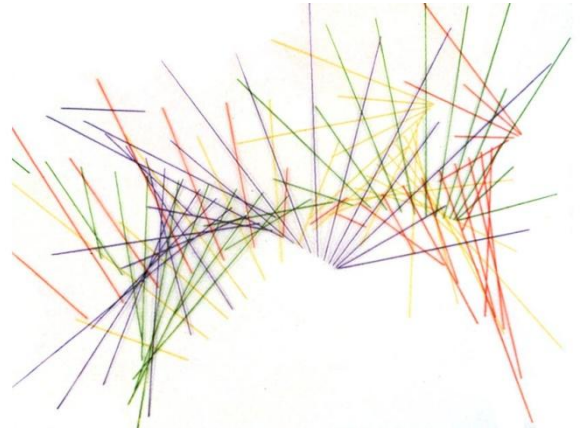


Lesbrief Algoritmisch tekenen

Inleiding

Algoritmisch tekenen is tekeningen maken volgens simpele regels, die tot een complex resultaat leiden. Je kunt het doen door een computer te programmeren, maar ook zelf met potlood en papier. In deze workshop maken we kennis met beide vormen, die je analoog en digitaal zou kunnen noemen, en kijken we naar het verschil tussen het geven van een opdracht geven aan een mens en aan een computer. We maken kennis met de basisprincipes van programmeren en schrijven ons eerste computerprogramma.

Algoritmisch tekenen wordt op het WDKA Interaction Station als open aanbod aangeboden. In deze workshop vliegen we in vogelvlucht door het normaal 8 uur lange programma heen.



Lesmiddelen

- Pennen, stiften, potloden
- Geodriehoeken
- Papier
- Werkblad met conditional design instructies
- Speciale analoge instructieblokjes
- Overhead projector
- Computers met python + libraries
- 2 Plotters

Conditional Design

Het maken van grafische patronen op basis van regels en geometrie is al erg oud, zo zien we het bijvoorbeeld terug in oude islamitische kunst (afb. 1). Een recentere ontwikkeling is de interesse voor “Conditional Design”, een methode waarin vaste voorwaarden en regels leiden tot een onverwacht resultaat. Het recent verschenen “Workbook” is een vaak aangehaald voorbeeld van deze stroming (afb 2.). Een pionier op dit gebied was is Sol Lewitt, een kunstenaar die diverse werken in de vorm van “instructions” maakte (afb 3).



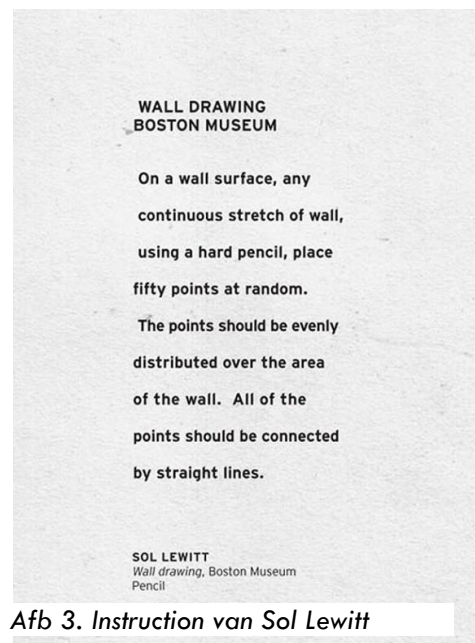
Afb 1. Tegels in het darb-e imam heiligdom

Opdracht 1 – Enkele korte Conditional Design oefeningen (10 min.)

Op A4-tje krijgen we enkele korte instructies, waarvan ons gevraagd wordt die uit te voeren. Op eventuele aanvullende vragen geeft de instructeur bewust geen antwoord. Na afloop vergelijken we de resultaten.



Afb 2. Conditional Design workbook



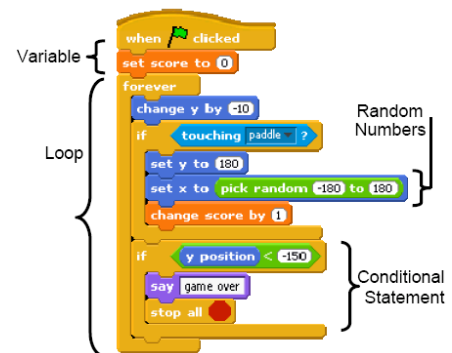
Afb 3. Instructie van Sol Lewitt

Programmeren

We zien dat bij de uitvoering van opdracht 1 er verschillen ontstaan; niet alleen door de willekeur die in de regels besloten zit, maar ook door de verschillende keuzes in de manier van hoe men te werk gaat, die van persoon tot persoon verschillen. Bij de instructie “trek een lijn” pakt de één zijn lineaal terwijl de ander de uit de losse pols tekent, wanneer hier van te voren niets duidelijk over is aangegeven. Dit verschilt met het geven van opdrachten aan een computer: deze kan niets “zelf” invullen, en dus zal hem alles moeten worden verteld. Gelukkig zijn veel opdrachten al beschreven in zogenaamde “libraries”. Zo ontstaat het een verschil tussen zogenaamde hoge en lagere programmeertalen.



Afb 5. LOGO (1967)



Afb 6. Scratch (scratch.mit.edu)

Opdracht 2 - Analooq programmeren (40 min.)

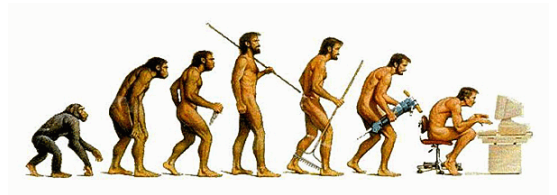
Voordat we achter de computer gaan zitten, gaan we eerst programmeren “met onze handen”, met speciaal hiervoor ontworpen instructie-blokjes.

Na een introductie in de werking van de verschillende blokjes, en enkele voorbeelden (20 min.), worden we uitgedaagd zelf letterlijk een programma in elkaar te zetten. (10 min.). Vervolgens wisselen we de programma’s door, en moeten we zelf “voor computertje spelen” door elkaars programma uit te voeren met potlood en papier. (10 min)

Het met elkaar en met de handen bezig zijn in deze opdracht heeft zo zijn voordelen, het voorkomt dat iedereen gelijk “in zijn eigen scherm verdwijnt”. De vorm van programmeren heeft zelf ook educatieve roots: we baseren onze teken-programmeertaal op het turtle-paradigma dat eind jaren 60 werd ontwikkeld om studenten de LOGO programmeertaal te leren (afb 4). Het mooie van deze methode is dat ieder uitgevoerd commando gelijk een zichtbare uitwerking heeft. Daarnaast proberen we de structuur inzichtelijk te maken door te werken met in elkaar passende modules, geïnspireerd op de door MIT ontwikkelde educatieve programmeeromgeving Scratch (afb 5)

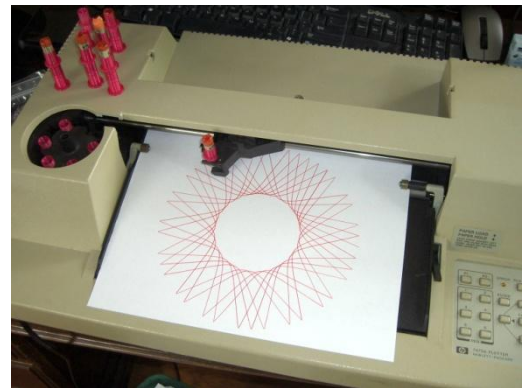
Van analoog naar digitaal...

Zoals op zoveel gebieden in ons leven heeft de computer ook op visueel gebied voor een revolutie gezorgd. Door ons tijdrovend werk uit handen te nemen kunnen we veel sneller een ontwerp maken, en bijvoorbeeld makkelijk zien welke invloed kleine veranderingen invloed hebben wanneer ze worden doorgevoerd in een groot systeem. We kunnen de ontwerpen daarnaast met een precisie uitvoeren die met de hand nooit te bereiken is.



...en weer terug naar analoog.

Tegelijkertijd zien ook een tegenbeweging op deze technische perfectie, een liefde voor het imperfecte, het organische, het analoge. Dit heeft bijvoorbeeld geleid tot een revival van de ouderwetse *plotter*, een voorloper van de moderne printer, waarin met een fysieke pen over het papier kan worden bewogen om tekeningen te maken (afb 6.). Terwijl ze in de jaren 90 door bedrijven massaal werden weggedaan, worden ze nu herontdekt door kunstenaars.



Afb 6. HP 7475A Plotter (1983)

Opdracht 3 - Programmeren en plotten met Python (40 min.)

De programma's die we met de blokjes hebben gemaakt, kunnen één op één worden overgezet naar een zelfgeschreven python-script. Python is een bekende script-taal die wijdverbreid wordt gebruikt voor allerlei ICT-toepassingen.

Door een computer te programmeren, wordt ons het tijdrovende tekenen uit handen genomen. We kunnen dingen nu makkelijk vaak herhalen, en aanpassen doen met variabelen om te zien wat voor effect dit heeft op de tekening. Zo schaven we ons programma bij en veranderen we ons patroon.

Wanneer we tevreden zijn, sturen we ons programma met hulp van de instructeur naar de plotter, en laten het zo op een fysieke manier uitvoeren. Zo is de cirkel rond en zijn we weer terug bij het pen en papier waar we mee begonnen zijn.