

# #KLOOIENMETCOMPUTERS

ARNOUT VAN KEMPEN OVER ROMMELEN IN EEN DIGITALE WERELD

In 1970 bedacht de Britse wiskundige John Conway een 'spel' dat Life heet. Het bestaat uit een oneindig raster van punten, met enkele extreem simpele regels. Vanaf een startpositie, waarbij je een aantal punten aan zet, ofwel levend, wordt iedere generatie ieder punt opnieuw berekend. Iedere levende cel met twee of drie levende buurcellen blijft leven. Bij vier of meer levende buurcellen en bij minder dan twee levende buurcellen, gaat een levende cel dood. Als een dode cel exact drie levende buurcellen heeft, wordt deze cel ook levend. In een generatie worden alle cellen tegelijk herberekend en weergegeven. Daarna begin de volgende generatie. Je kan hier reuze filosofisch over doen, maar wat in de praktijk vooral leuk bleek:

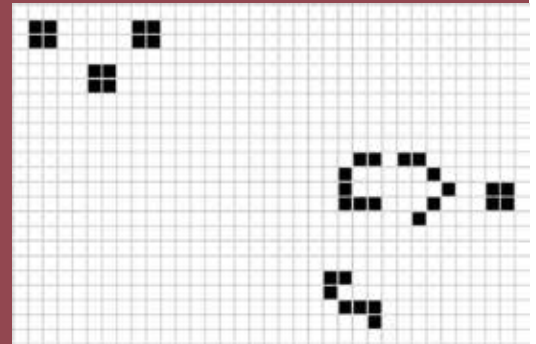
- 1 Celgroepen, ook wel organismes genoemd, ontwerpen met eigenschappen zoals continuïteit, verplaatsing explosies en implosies en zelfs voortplanting.
- 2 Zelf een programma schrijven om Life mee te spelen.

Als student deed ik ooit mee aan een programmeerwedstrijd van het blad PCM, een blad voor computerhobbyisten. De opdracht: schrijf een programma om *Life* mee te spelen. Het was in de tijd dat Windows begon op te

komen en mensen snelle 386 en 486 pc's hadden. Alle inzendingen werkten dan ook met Windows, behalve de mijne. Ik had een wat oudere 286 met MS/DOS, maar ik vond vooral dat *Life* pas leuk is als het zichtbare raster groot is, en generaties wisselen met een snelheid zonder zichtbare vertraging. Met een echt groot raster was die snelheid, zeker onder Windows, nauwelijks te doen.

Ik koos niet alleen om Windows te negeren, ik koos ook voor de snelst mogelijke programmeertalen die op dat moment mogelijk waren. Waar veel inzenders gebruikmaakten van Pascal-varianten en Visual Basic, zo ongeveer het Python van vandaag, koos ik er voor om alle functionaliteit in C te schrijven en de generatieberekening zelfs in Assembler. Mijn inzending miste de typische mooie dingen van Windows, met menustructuren, een grafisch aantrekkelijk uiterlijk en mogelijkheden van multitasking. Alle andere inzendingen hadden echter een trage generatieberekening en een in mijn ogen absurd klein zichtbaar raster. Sterker, ik vermoed dat mijn versie aanzienlijk dichterbij het idee van Conway kwam.

Natuurlijk werd mijn inzending door de redactie in het blad volledig afgebrand wegens niet mooi genoeg en werd de krankzinnig hoge snelheid van mijn inzending genegeerd. Ik kreeg nog wel enige



genoegdoening via mijn opleiding. Daar leverde ik mijn project ook in, kreeg een 10 als eindcijfer en ik hoefde geen les meer te volgen.

Als u nu ook eens met *Life* aan de slag wilt, zoek het dan eerst eens op in uw appstore. Android zal ongetwijfeld varianten kennen, op de iPhone raad ik *Game of Life* aan. En als u zelf aan de slag wilt, zeker doen. Het is niet krankzinnig moeilijk en wel heel bevredigend om een dergelijk project tot een goed einde te brengen. Persoonlijk zou ik met de huidige stand van hardware niet langer voor Assembler kiezen. Op moderne computers is C meer dan snel genoeg en zelfs het vrij trage Python komt een heel eind.

Van Java ben ik persoonlijk geen fan. En als je twijfelt tussen Duits of Java, zou ik Duits echt belangrijker vinden, zeg ik met een knipoog naar Lena van Kruijdsijk. ←