

Grafische programmeertalen

Bij programmeren denken veel mensen aan moeilijke, onbegrijpelijke en onleesbare tekens en codes. De eerste gedachte die vaak opkomt is: Dit snap ik niet en kan ik niet. Een logisch gevolg is dat ze het al helemaal niet aanbieden aan kinderen.

Het beeld van programmeertaal als onleesbare, onbegrijpelijke code komt vooral uit de jaren zeventig tot en met negentig, toen de wereld kennismakte met de computer. Windows en tablets bestonden nog niet, en om een programma te starten moest je commando's invoeren achter een knipperend wit streepje. Programmeren was iets voor heel slimme mensen die de hele dag aan het 'code kloppen' waren. Zij konden bijvoorbeeld door middel van onderstaande code *Hallo, wereld!* op het scherm laten verschijnen met behulp van de programmeertaal C++.

```
#include <iostream>

int main()
{
    std::cout << "Hallo, wereld!";
}
```

Tegenwoordig bestaan er grafische programmeertalen. De meest bekende is Scratch, een programmeertaal en -omgeving ontwikkeld voor kinderen vanaf 8 jaar. In Scratch ziet bovenstaande code er zo uit:



Net Lego

Grafische programmeertalen ontwikkelen het denken als programmeur (*computational thinking*), maar leiden een kind niet af met Engelstalige commando's en strikte syntax. Door een paar blokken met commando's aan elkaar te klikken – het is net Lego – kan een kind een kat over het scherm laten lopen. Hij heeft daardoor direct een succeservaring.

Programmeren kun je leren door code van anderen na te maken en aan te passen; afkijken is zelfs verplicht! Bij tekstuele programmeertalen is het dan wel handig als je snapt wat er staat, want anders ben je een kunstje aan het nadoen. Aangezien programmeertalen op het Engels gebaseerd zijn, betekent dit dat Nederlandstalige kinderen, uitzonderingen daargelaten, pas rond een jaar of twaalf serieus met programmeren kunnen beginnen. Terwijl Engelstalige kinderen dit al vanaf een jaar of acht doen.

Door kinderen op jonge leeftijd te leren programmeren, breng je ze in aanraking met vaardigheden waar ze in de toekomst (en nu ook al) profijt van hebben, zoals creatief en logisch denken, structureren, analyseren, probleemoplossend vermogen en samenwerken. De meeste kinderen vinden het heerlijk om scheppend bezig te zijn, terwijl ze in het huidige (basis)onderwijs vooral leren hoe ze de computer moeten bedienen in plaats van besturen.

De reclame van Planet Internet van tien jaar geleden ('Ga met je muis naar de rechterbovenhoek') is ook nu nog actueel. Programmeren leert kinderen juist maken/creëren/scheppen in plaats van consume-

door Sjoerd Dirk Meijer

ren. Dit betekent dat je al vroeg kunt – en misschien wel moet – beginnen met het aanbieden van programmeren.

Spelend programmeren

De laatste jaren is er ontzettend leuk en leerzaam programmeermateriaal op de markt gekomen voor kinderen vanaf vier jaar. Met het bordspel (dus niet digitaal!) Robot Turtles leren kinderen onder andere denken in stappen, volgorde en vooruit kijken. Ook de Primo-speelset leert kinderen de basisprincipes van het programmeren, met behulp van een houten robotje genaamd Cubetto en een programmeerbord. Met verschillende soorten blokjes (rood voor '1 stap vooruit', blauw voor 'draai naar links', geel voor 'draai naar rechts') programmeren kinderen de route die Cubetto gaat afleggen. Terwijl ze spelen, zijn kinderen met deze materialen op een ongedwongen manier aan het programmeren.



Voor kinderen vanaf vijf jaar is de gratis iPad- en Android-app 'ScratchJr' heel geschikt. Dit is het jonge broertje van Scratch, en ook hier wordt geprogrammeerd met blokken. Op de blokken staan icoontjes en geen tekst, waardoor de app geschikt is voor kinderen die niet of nog maar net kunnen lezen.

Voor kinderen vanaf acht jaar biedt Scratch een zeer geschikte omgeving om te leren programmeren. Scratch is geschikt voor kinderen die als doel hebben een creatief project te maken, waarbij programmeren als middel wordt gebruikt om het te realiseren, maar ook voor kinderen die het programmeren onder de knie willen krijgen, waarbij creativiteit als middel wordt gebruikt om dit vorm te geven. Scratch doet bijna niet onder voor een tekstuele programmeertaal en is daardoor ook voor pubers nog zeer geschikt om te leren programmeren.

Vaardigheden ontwikkelen

Het is uiteraard niet de bedoeling om alle kinderen op te leiden tot programmeur. Of je nu wel of niet kiest voor een informatica- of technische opleiding, in elk beroep kunnen analyseren, structureren, problemen oplossen en logisch denken voordeel opleveren.

Bij programmeurs denken veel mensen aan hoogbegaafde, sociaal niet-vaardige nerds, die alleen maar in computertaal kunnen communiceren. Het stereotiepe beeld is een *nerdy* man die alleen maar achter een beeldscherm vol code zit. Hierdoor denken veel ouders dat programmeren niet past bij hun zoon – en al helemaal niet bij hun dochter. De werkelijkheid is gelukkig anders. Om te leren programmeren zijn interesse en inzet het allerbelangrijkste. Code doet vaak iets anders dan de programmeur had bedacht. Het vraagt dan soms veel doorzettingsvermogen om dat ene kleine foutje in de code te ontdekken. Een kind zal de inzet daarvoor tonen als het gedreven is en/of het leuk vindt. Jongens hoeven niet per definitie beter te kunnen (leren) programmeren dan meisjes; de allereerste programmeurs waren vrouwen! En een hoge intelligentie en een goed gevoel voor wiskunde zijn zeker voordelen, maar geen voorwaarde.



Toch past leren programmeren bij veel (vermoedelijk) hoogbegaafde kinderen en helpt het ze bij hun specifieke ontwikkeling. Hoogbegaafde kinderen hebben behoefte aan uitdagende opdrachten, waarin ruimte is voor exploratie en eigen inbreng. Bij het aanleren van programmeren kan dit door het maken van een uitdagend computerprogramma, waarbij de basisstof in één keer behandeld wordt in plaats van stap voor stap. Programmeren komt tegemoet aan de behoefte om te scheppen en creatief bezig te zijn. Met programmeren kun je de computer naar je hand zetten en hem laten doen wat jij bedacht hebt. Hoogbegaafden vinden het vaak heerlijk om de gekste, vreemdste, creatiefste dingen te bedenken, maar nog leuker is het om die tot leven te brengen. Als het computerprogramma doet wat het kind bedacht heeft, dan is het goed gemaakt – ook al staan alle figuren in het project op de kop of hebben ze heel vreemde kleuren.

Maar vaak doet een programma het anders dan dat het kind bedacht heeft. En dat is een hoogbegaafde niet gewend! Zeker faal-

Sjoerd Dirk Meijer was leraar basisonderwijs en functioneel beheerder bij meerdere overheidsinstellingen. Zijn kennis van programmeren en het (basis)onderwijs combineert hij in zijn bedrijf iQMaak, waarin hij programmeercursussen en -workshops organiseert voor kinderen en leraren, en scholen ondersteunt bij het integreren van programmeren in het onderwijs. Voor (vermoedelijk) hoogbegaafde kinderen organiseert hij doelgroepgerichte cursussen en workshops.
www.iQMaak.nl

angstige kinderen leren veel van programmeren: fouten maken is niet erg. Het moet zelfs! Omdat een kind zelf kan bedenken wat het gaat maken, zit programmeren bij een hoogbegaafde meestal op het frustratieniveau. Als gevolg hiervan ervaren de kinderen ook dat doorzetten beloond wordt: na het oplossen van de 'fouten' (eventueel met wat hulp) is het project af en zijn ze trots!

En dat kan mooie gevolgen hebben. Een 5-jarige met een *bore-out* vertelde me voorafgaand aan een les dat hij nog zes dagen nodig had om beter te worden. Na de les vroeg ik hem of hij nog steeds zes dagen nodig had. Zijn reactie raakte me: 'Nee, dat is nu wel een paar dagen minder geworden.' ■