

# Werken met een robot is meer dan leuk spelen

**In de onderbouw aan de slag met programmeren?** Op veel scholen zien we dat robots met dat doel de klas in worden gehaald. Bijvoorbeeld een Bee-Bot of Robotmuis. Met deze robots breng je kinderen op een speelse manier in aanraking met programmeren. Maar hoe zorg je ervoor dat het meer is dan leuk spelen?

**B**edrijven, onderwijsprofessionals en onderzoekers zijn het erover eens dat *Computational Thinking* een belangrijke vaardigheid is waarvan kinderen in de toekomst gebruik zullen maken. In het onderwijs van de toekomst zal deze vaardigheid naar verwachting dan ook een belangrijke plaats gaan innemen. *Computational Thinking* is door SLO omschreven als 'het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen'. Programmeren is een manier waarop kinderen aan hun *Computational Thinking* vaardigheden kunnen werken. Maar hoe breng je kinderen in de onderbouw van het basisonderwijs in aanraking met programmeren? En hoe kun je jonge kinderen gericht helpen bij het ontwikkelen van programmeervaardigheden? Om die vraag te kunnen beantwoorden, moeten we eerst inzoomen op wat programmeervaardigheden zijn.

## Programmeren in de onderbouw

Zowel nationaal als internationaal is er momenteel veel aandacht voor brede (of toekomstgerichte) vaardigheden. In Nederland is recent in het kader van *Curriculum.nu* een generiek model ontwikkeld waarin relevante vaardigheden worden samengebracht<sup>2</sup>. Het model onderscheidt drie groepen vaardigheden: manieren van denken en handelen, manieren van omgaan met anderen en manieren van jezelf kennen. Programmeren vereist creatief, kritisch en probleemoplossend denken en handelen. Ze vallen in de eerste groep. Het is belangrijk dat kinderen deze vaardigheden ontwikkelen. Maar wat is probleemoplossend denken bij programmeren? En wat is creatief of kritisch handelen? Kennisnet noemt zes bouwstenen. Vanuit deze bouwstenen kun je in de onderbouw met programmeren aan de slag.

Programmeren in de onderbouw: 6 bouwstenen <sup>3</sup>	
<b>Algoritmes</b>	Een simpel algoritme kunnen maken voor een concrete situatie met een vooraf bepaalde set instructies.
<b>Decompositie</b>	Een probleem kunnen opdelen in deelactiviteiten en deze in een logische volgorde kunnen plaatsen.
<b>Patronen</b>	Patronen in bijvoorbeeld model, kleur of vorm kunnen herkennen in een activiteit of stappenplan.
<b>Herhaling</b>	Begrijpen dat herhalingen activiteiten zijn die je automatisch meerdere keren kunt laten uitvoeren door een computer.
<b>Fouten</b>	Begrijpen wat debuggen betekent en met een aantal strategieën simpele (proces)fouten kunnen oplossen.
<b>Voorwaarden</b>	Begrijpen wat het statement 'Als ... dan ...' betekent en op basis van vooraf bepaalde tekens en acties met voorwaarden kunnen werken.

## Leren met een robot

Er zijn veel manieren om te oefenen met programmeren. Op de computer, met of zonder lesboek, maar een populaire manier is door te werken met robots. Door veel leerkrachten wordt het als voordeel gezien dat je met een robot bezig bent met een digitale vaardigheid, terwijl je niet per se achter een computerscherm hoeft te zitten. Dat maakt het makkelijker te integreren in het klaslokaal. Veel basisscholen experimenteren er dan ook al mee. Het werken met een robot kan zorgen voor een aangename leerervaring én leiden tot betere leerprestaties. Een win-win situatie dus. Maar niet als het alléén leuk spelen is. Samen met basisscholen heeft CitoLab, de onderzoeks- en innovatieafdeling van Stichting Cito, uitgezocht hoe je kinderen in de onderbouw van het basisonderwijs opbrengstgericht kunt laten werken met de robot-programmeermuis. Daarbij is gebruikgemaakt van het speelveld in *Figuur 1*.



Figuur 1. Speelveld 'De dierentuin'

Om het speelveld en de robot opbrengstgericht te gebruiken in de les kun je het volgende doen:

### A - Geef vooraf instructie

Leren begint met een goede instructie. Sommige kinderen kennen de robot al van thuis en kunnen er direct mee werken. Andere kinderen moeten eerst even oefenen. Vertel in ieder geval dat (1) de robot precies doet wat jij zegt, (2) de pijlknoppen links en rechts de robot op zijn

plek laten draaien, en (3) de robot alles onthoudt totdat je de wisknop indrukt. Laat kinderen zelf de knoppen uitproberen en geef oefenopdrachten om te kijken of ze de knoppen op de robot kunnen bedienen. Vraag bijvoorbeeld om de robot twee vakjes vooruit of opzij te laten rijden. Doe het samen als het niet lukt.

### B - Kies een leerdoel en opdracht

Je kunt de robot allerlei soorten opdrachten geven. De opdrachten kun je willekeurig kiezen, maar beter is het om de opdracht gericht te kiezen bij het leerdoel waar je met het kind aan wilt werken. Op basis van de zes bouwstenen kun je een keuze maken. Er is nagedacht over de volgorde van de bouwstenen en de bijbehorende voorbeeldopdrachten. De opdrachten nemen toe in complexiteit en abstractie.

#### Voorbeeld programmeeropdrachten bij het speelveld 'Dierentuin'

<b>Algoritmes</b>	Kun jij de robot helpen om bij <de zeehond> te komen?
<b>Decompositie</b>	Hoe komt de robot via <de leeuw> snel bij <de olifant>?
<b>Patronen</b>	Vergelijk de knoppen en de pijlkaarten. Zie je een patroon?
<b>Herhaling</b>	Als je de robot dezelfde route nogmaals laat lopen, waar kom je dan?
<b>Fouten</b>	De robot moet naar <de vos>. Waar gaat het mis?
<b>Voorwaarden</b>	Kies <3> keer <vooruit - draai linksom - vooruit>. Waar komt de robot? En als je <draai linksom> verandert in <draai rechtsom>?

### C - Observeer en schrijf op wat je ziet

Via gerichte observatie kun je allerlei dingen te weten komen over de programmeervaardigheid van kinderen. Het is gemakkelijk om bij te houden of een kind de opdracht succesvol weet uit te voeren. Maar je kunt ook bijhouden hoeveel pogingen er nodig zijn, hoe op de opdracht 'opgelost' wordt en welke fouten daarbij gemaakt worden. Bij oplossingsstrategieën kun je bijvoorbeeld denken aan *Denkt route uit in het hoofd*, *Tekent de route uit op het bord of Begint direct met programmeren*. Bij fouten kan het gaan om *Vakje overgeslagen*, *Draai gemist*, *Verkeerd geteld* etc. Op basis van deze informatie kun je vaststellen bij welke bouwsteen het kind staat. Daarnaast kun je de informatie gebruiken om het kind op maat te begeleiden naar de volgende bouwsteen. Precies wat opbrengstgericht werken inhoudt. En precies wat ervoor zorgt dat een robot meer wordt dan alleen leuk spelen.

### Spelvarianties

Wij hebben een dierentuin als speelveld gebruikt. Het speelveld bestaat uit  $5 \times 5 = 25$  vakjes. De robotmuis werd op een plek op het speelveld gezet. Aan kinderen werd gevraagd om de robotmuis te programmeren om bij een bepaald dier op het speelveld uit te komen. Kinderen waren enthousiast! En de deelnemende scholen ook. Maar uiteraard is er meer mogelijk. Spelvarianties zijn eenvoudig online te vinden of zelf te verzinnen. Andere speelvelden zijn bijvoorbeeld te downloaden via internet. Ook kun je de kinderen een blanco speelveld geven en een eigen ontwerp laten maken. In *Figuur 2* kun je een voorbeeld zien. Obstakels toevoegen is ook een leuk idee. En niet alleen dat. Je maakt de opdrachten er gelijk ook moeilijker mee. Daarnaast kun je:

- Kinderen zelf een verhaal met bijhorende route bij het speelveld laten bedenken;
- Met twee robots op hetzelfde speelveld werken en kinderen laten samenwerken;
- Eigen regels toevoegen aan het spel. Bijvoorbeeld: 'Je mag niet over het vakje met ...'
- Het speelveld uitbreiden met meer vakjes om opdrachten complexer te maken;
- Aan kinderen in hogere leerjaren vragen om een speelveld te maken en de robot uit te leggen aan de kinderen in de onderbouw.



Figuur 2. Een door leerlingen gemaakt speelveld met obstakels

#### Manieren om hulp te geven

Activiteit	Effect
Laat het kind 'doelkaartjes' neerleggen. Bijvoorbeeld een kaartje met het cijfer 1 op het vakje met <de ijsbeer> en het cijfer 2 op <de leeuw>.	Ontlasten van het werkgeheugen. Onthouden van het einddoel is niet meer nodig, alle aandacht kan uitgaan naar het programmeren.
Laat het kind de route visualiseren door 'pijlkaartjes' op een rij te leggen en daarna deze route intoetsen.	Door de visualisatie kunnen het kind en jijzelf beter zien welke denkfouten gemaakt worden.
Zit er een fout in de route? Laat het kind aanwijzen bij welke pijl(en) het misging en pas de 'pijlkaarten' aan.	Het kind kan makkelijker over zijn route nadenken: hij kan terug redeneren wat er fout ging.
Gebruik 'pijlkaarten' voor toepassing van de placeholder-techniek: voor elk pijlkaartje dat het kind neerlegt, verplaats jij met de hand de robot op het speelveld.	Door een actie direct op het speelveld te visualiseren, ziet het kind welk gevolg zijn denkstap heeft.

### Tot slot

Werken met de robot-programmeermuis of een andere robot is een leuke activiteit. Dat staat voorop en dat moet zo blijven. Maar door opdrachten beredeneerd te kiezen en gericht te observeren kun je kinderen écht helpen om nóg beter te worden in programmeren. Dat kan spelenderwijs. Jonge kinderen laten in vertrouwde situaties vaak bijna als vanzelf allerlei vaardigheden zien. Dan voegt observatie echt wat toe. Je komt te weten waar het kind staat in het leerproces. En het vertelt je hoe je het kind verder kunt helpen in het leren. Dan wordt het besturen van robots méér dan leuk spelen. Wil jij ook gebruikmaken van de materialen die ontwikkeld zijn? Het speelveld en de hulp- en opdrachtkaarten kun je samen met een aantal 'Wijzers' gratis downloaden via [cito.nl/programmerenmetrobots](http://cito.nl/programmerenmetrobots). ■

#### Referenties

- 1 <http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking>.
- 2 <https://curriculum.nu/wp-content/uploads/2018/04/Rapport-Handreiking-bredewaardigheden.pdf>
- 3 [https://maken.wikiwijs.nl/74282/Programmeren\\_in\\_het\\_PO#!page-1843082](https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO#!page-1843082)