

## HOOFDSTUK 5: TAFELS OEFENEN

### TIJDSINDICATIE

- 70 minuten
- 80 minuten mét afspelen filmpje *Sandwich robot* (let op: misschien heb je dat in Hoofdstuk 1 al gedaan). Dit is de link <https://www.youtube.com/watch?v=wpC56gn79YQ&t=31s>
- 90 minuten als je leerlingen laat nadenken over het opstellen van het algoritme (punt 1 op pag 45)


### VOORKENNIS

- Onderwerpen uit Hoofdstukken 1 t/m 4
- Opslaan, rugzak, initiëren

### DOELEN TEN BEHOEVE VAN DE LEERLINGEN

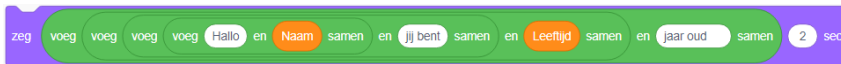
- Tafelsommen maken. Daarbij gaat het de schrijvers van het boek om het proces. Maar misschien gaat het jouw klas/jouw school wel om het oefenen met tafels. Bij dit spel is sprake van integratie: programmeren wordt gecombineerd met een nuttige rekentool.
- Functie 'samenvoegen'
- Random: willekeurig. De computer willekeurige getallen laten kiezen. Toepassing en nut worden verderop toegelicht.
- Werken met geluiden, geluid opnemen en deze in je programma gebruiken.

### LEERLIJN

- Abstractie, in de meest ruime zin. Kennisnet zegt daar o.a. over: *Leerling kan op een zodanige manier problemen formuleren dat het mogelijk wordt om het probleem op te lossen door gebruik van een computer of ander gereedschap.*
- Functie. Functies is een van de programmeertools uit het Blokkenpalet. Wij gebruiken de functie  om te checken of het door de gebruiker ingegeven antwoord overeenkomt met het juiste antwoord van de keer-som: in het linker deel komt de som te staan, het rechter deel bevat het door de gebruiker gegeven antwoord. Vervolgens checkt computer of de bewering waar is: of het goede antwoord is gegeven. Zo ja, dan volgt er zús, zo nee, dan volgt er zó.

### LESVERLOOP, LES SUGGESTIES

- 'Wie ben je?' is een opwarmertje. Het samenvoegen van teksten wordt geoefend, zeg maar: het aan elkaar plakken van stukken tekst en variabelen. Een computer kan geen zinnen samenstellen. De gebruiker plak stukken van een zin en waarden van een variabele aan elkaar. De gebruiker heeft deze waarde ingevoerd, bijvoorbeeld: *naam* of *leeftijd*. Voorbeeld:



- 'Tafelsommen programmeren' heeft uitleg nodig. Het lastigste in deze paragraaf is het opstellen van het algoritme. Denk maar terug aan de sandwich robot, die is ook lastig te programmeren! Je kúnt er voor kiezen bij punt 2 op pag 45 te beginnen. In dat geval ben je aan het ná bouwen en sla je het deel dat leerlingen nadenken waarbij ze het algoritme opstellen (wat voor nu echt heel moeilijk is) over.

Echt programmeren is natuurlijk dat de programmeur het algoritme samenstelt. De ervaring leert dat leerlingen het stappenplan 'hoe bied ik willekeurige tafelsommen aan de gebruiker aan, waarbij ook nog wordt gecontroleerd of het antwoord goed is', niet eenduidig weten te verwoorden. Dat is niet gek, want echt programmeren kan pas als je het hele boek uit hebt en dan nog is het opstellen van algoritmes super moeilijk. Vergelijk het met rekenen: dat leer je ook niet in 7 lessen. Misschien is voor jouw klas nu het bouwen en het begrijpen wat er gebeurt prima.

- Als je een uitdaging aandurft: laat leerlingen het met pen en papier stap voor stap uitwerken, dus unplugged en daarna programmeren. Nogmaals: succes niet verzekerd! Dit is punt 1 op pag 45. Voorbeeld:



*Persoon1 is de computer, persoon2 is de speler. Speel het stap voor stap. Neem bijvoorbeeld de tafel van 3 en doe 4 sommen. Op deze manier blijft het concreet, dat is praktisch. Het kan er zo uitzien: Persoon1 zegt "welke tafel wil je oefenen?". Persoon2 zegt "3". Persoon1 zegt "hoeveel sommen wil je oefenen?". Persoon2 zegt "4". Denk nu na hoe de computer een som maakt, bijvoorbeeld:  $7 \times 3$ . Weet je dat de 7 hier willekeurig gekozen is door de computer? Speler 2 geeft antwoord. Dat antwoord kan goed of fout zijn. En zo ga je verder. Houd steeds in gedachten dat je in Scratch-taal moet denken.*

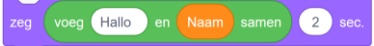
- 'Uitdaging: score bijhouden'. Snelle leerlingen kun je nog het aantal goed en fout beantwoorde vragen laten tonen. Toon ook het percentage vragen dat goed is beantwoord. Op deze manier krijgt programmeren toegevoegde waarde ten behoeve van rekenen en is er sprake van vakintegratie.

## EVALUATIE

- Bespreek het unplugged gedeelte met de klas. Of: laat een algoritme van een leerling door een andere uitvoeren, net zoals de Sandwichrobot eigenlijk.
- Stukken tekst aan elkaar koppelen met de samenvoeg-functie vergt nauwkeurig werken, zeker als er vaker dan één keer wordt samengevoegd.
- Als het opstellen met pen en papier van het tafelsom algoritme lastig blijkt: het boek werkt het stap voor stap uit. Dus mochten leerlingen er met pen en papier niet, of lastig, uitkomen: volg de uitwerkingen van paragraaf 'Tafelsommen programmeren' vanaf punt 2 ipv punt 1!

## ACHTERGRONDINFORMATIE

Na het script  volgt altijd het script . Daarmee geef je de variabele de waarde mee die je hebt ingetoetst. De variabele *naam* krijgt de waarde van wat de gebruiker heeft ingetoetst. Dit wordt vaak gebruikt.

De 'voeg samen' functie binnen onze zeg-boodschap  plakt stukken aan elkaar, in dit geval: het woord 'Hallo' en daar achter de waarde van de variabele *naam*. Dat betreft hier de naam die de gebruiker heeft ingevoerd. Je kunt zoveel stukken tekst en waarden van variabelen samenvoegen als je wilt.

## MEER INFORMATIE

- Abstractie is het zevende onderwerp beschreven in de leerlijn programmeren. Zie: [https://maken.wikiwijs.nl/74282/Programmeren\\_in\\_het\\_PO#!page-1838213](https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO#!page-1838213)
- Functie is het achtste onderwerp beschreven in de leerlijn programmeren. Zie: [https://maken.wikiwijs.nl/74282/Programmeren\\_in\\_het\\_PO#!page-1838219](https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO#!page-1838219)