

Bachelorproef  
Professionele Opleidingen  
Studiegebied Onderwijs

Academiejaar 2017-2018

# Programmeren, de vaardigheid van de toekomst

Hoe kun je programmeren integreren in de kleuterklas en wat zijn de voordelen hiervan?

Bachelorproef aangeboden door  
**Ikram Elhihi**

tot het behalen van de graad van

**Bachelor in het Onderwijs: Opleiding Kleuteronderwijs**

Interne begeleider: **Jaantje Verbruggen**

Bachelorproef  
Professionele Opleidingen  
Studiegebied Onderwijs

Academiejaar 2017-2018

# Programmeren, de vaardigheid van de toekomst

Hoe kun je programmeren integreren in de kleuterklas en wat zijn de voordelen hiervan?

Bachelorproef aangeboden door

**Ikram Elhihi**

tot het behalen van de graad van

**Bachelor in het Onderwijs: Opleiding Kleuteronderwijs**

Interne begeleider: **Jaantje Verbruggen**



# Woord vooraf

Als studente kleuteronderwijs aan de Hogeschool Odisee te Brussel, heb ik veel plezier beleefd aan het schrijven van deze bachelorproef. Ik heb twee onderwerpen, kinderen en programmeren, die op het eerste gezicht niets met elkaar te maken hebben, kunnen assembleren tot één geheel.

Ik wil heel graag mijn promotor, Jaantje Verbruggen, bedanken voor haar hulp, tips en begeleiding. Ook mijn mentor Femke Van Dewalle wil ik bedanken voor haar vertrouwen. Zij heeft mij veel zaken laten uitproberen in haar klas.

Tot slot wil ik mijn echtgenoot en software ingenieur, Abdel, in het bijzonder bedanken voor zijn expertise en morele steun tijdens dit proces.

Ik wens u veel leesplezier toe.

Ikram El Hihi

5 Juni 2018

## Inhoudstafel

2.	Programmeren en kleuters.....	6
3.	Programmeren in ons dagelijks leven.....	7
	3.1 Programmeertalen.....	7
4.	Programmeren met kleuters.....	8
	4.1 Waarom zouden kleuters moeten kunnen programmeren?.....	8
	4.2 Algoritmisch denken.....	9
	4.3 Computationeel denken.....	10
5.	Hoe populair is programmeren momenteel in het onderwijs?.....	13
	5.1 Resultaten van de enquête.....	14
6.	Programmeren op kleuterniveau.....	17
	6.1 Bestaande robots en handleidingen.....	17
	6.2 De Bee-Bot.....	17
	6.3 Jack the robot mouse:.....	18
	6.4 Scratch.....	19
	6.5 Programmeren voor kinderen.....	20
7.	Eigen activiteiten rond programmeren :.....	22
8.	Het thema robots.....	23
	8.1 De kleuters.....	23
	8.2 Samenwerking met een software ingenieur.....	23
	8.3 Opbouw van de week.....	24
	8.4 Voorbeeld wekschema en ideeën.....	24
	8.5 Losse brainstorm.....	25
	8.6 Mindmap per ontwikkelingsdomein.....	25
	8.9 Voorbeeld wekschema.....	27
9.	Aan de slag in de klas.....	28
10.	Alles in een notendop.....	31
11.	Bronvermelding.....	33

## **1. Programmeren en kleuters**

Programmatie, software, algoritme... allemaal woorden die niet meteen gelinkt worden aan kleuters. Het lijkt alsof deze woorden niets met elkaar te maken hebben, maar sinds een aantal jaren wordt de link tussen kleuters en programmatie steeds duidelijker.

Programmeren met en door kleuters is een recente techniek die steeds meer populair wordt in het onderwijs. Vooral in het lager onderwijs wordt deze techniek steeds meer gebruikt, maar ook voor het kleuteronderwijs bestaan er heel wat mogelijkheden.

In de huidige maatschappij heeft technologie een enorme impact. De kleuters van vandaag zijn de volwassenen van morgen, daarom is het belangrijk dat jonge kinderen op tijd en op een bewuste manier in contact worden gebracht met de nieuwste technologieën.

Programmeren op jonge leeftijd is een brede term, de nadruk ligt op creativiteit en het probleemoplossend denken. Op deze manier worden kinderen voorbereid om te leren omgaan met alle problemen uit het dagelijks leven.

In dit werk wordt onderzocht hoe programmeren geïntegreerd kan worden in een kleuterklas en wat de voordelen hiervan zijn. Bovendien hoort bij dit werk een handleiding vol ideeën om aan de slag te gaan met programmatie in een kleuterklas.

## **2. Programmeren in ons dagelijks leven**

Een gewone persoon, komt dagelijks in contact met programmatie en geprogrammeerde voorwerpen. Het is zo eenvoudig om een elektronisch apparaat te gebruiken, dat zelfs jonge kinderen deze apparaten kunnen gebruiken.

Zelden staan mensen stil bij het achterliggend mechanisme van deze apparaten. Een computer, een droogkast, een mixer, een telefoon, een drankautomaat en quasi alle voorwerpen die door de moderne mens dagelijks gebruikt worden, zijn op voorhand geprogrammeerd.

Deze toestellen worden geprogrammeerd door een programmeur of computeringenieur. In het jargon wordt er ook vaak gesproken van een softwareontwikkelaar.

De programmeur schrijft een reeks concrete instructies die de computer uitvoert. Het is een soort code die de computer begrijpt en omzet in acties. Daarom wordt programmeren ook wel 'coderen' genoemd.

Alvorens een toestel deze code begrijpt en kan uitvoeren, worden de instructies via een assembler, compiler of interpreter omgezet in machinecode. Via deze zeer ingewikkelde processen, wordt de code begrijpelijk voor het toestel en worden de instructies uitgevoerd.

Deze instructies worden dan ook gekoppeld aan een schakelaar, beweging of andere. Wanneer de gebruiker dan bijvoorbeeld de wasmachine wilt laten starten, moet de juiste knop worden ingedrukt en dan voert het toestel de instructies uit. (Programmeren, 2017)

### **Programmeertalen**

Programmeurs gebruiken verschillende talen om te programmeren. Elke taal heeft zijn eigen specifieke toepassing. De ene taal is bijvoorbeeld geschikt om wasmachines te programmeren en de andere taal is geschikt om spelletjes te ontwerpen.

Er ontstaan ook vaak nieuwe talen omdat de bestaande talen soms niet aansluiten bij het programmeren van een nieuw voorwerp of nieuwe techniek. Vaak zijn deze talen gebaseerd op bestaande programmeertalen en worden ze op bepaalde vlakken aangepast.

Hoe de verschillende talen van elkaar verschillen is iets zeer ingewikkeld en het vraagt veel ervaring om de verschillende talen te bergrijpen. Vaak gebruiken programmeurs ook de talen door elkaar om een bepaald doel te bereiken. (Programmeren, 2017)

Een programmeur kent gemiddeld twee tot vier programmeertalen. Hieronder even een lijst van de vijf meest bekende talen en hun toepassing.

Java is een van de populairste programmeertalen als het gaat om het bouwen van back-ends voor moderne webapplicaties. Dat betekent dat Java op de achtergrond allerlei taakjes uitvoert om de applicaties goed te laten werken.

Deze taal is ook één van de oudste talen en wordt zeer vaak gebruikt om Android apps te creëren.

Javascript is een derivaat van Java en wordt vooral gebruikt om websites interactief te maken en aan te passen aan de wensen en verwachtingen van de gebruikers.

C# is ook een zeer bekende programmeertaal. Deze taal wordt gebruikt voor het programmeren van Microsoftproducten. Het is een zeer brede taal, waarmee zowel websites als spelletjes en moderne webapplicaties geprogrammeerd kunnen worden.

C++ is een programmeertaal die het mogelijk maakt om rechtstreeks te communiceren met de hardware in je computer. Dit zorgt ervoor dat programmeurs het maximale uit een computer kunnen halen. Ze kunnen via C++ slimmer en effectiever programmeren.

Wanneer er gebruikt wordt gemaakt van een database, wordt PHP als programmeertaal gebruikt. PHP zorgt ervoor dat de data uit de database wordt gehaald en ook dat de data terug in de database worden geplaatst. Een artikel, bijvoorbeeld, kan dus op aanvraag (klik) getoond worden en terug verdwijnen.

Python is de breedste taal die er bestaat, het kan bijna alles. Toch wordt Python voornamelijk gebruikt bij het werken met grote hoeveelheden data. In vele universiteiten wordt deze taal gebruikt om de data van de onderzoekers te verzamelen. (Vos, 2015)

### **3. Programmeren met kleuters**

#### **Waarom zouden kleuters moeten kunnen programmeren?**

Programmatie heeft een invloed op de totale ontwikkeling van het kind. Wanneer kinderen in de klas programmeren, komen ze ook in contact met andere specifieke domeinen zoals: wiskunde, taal, wereldoriëntatie... Geïntegreerde activiteiten zijn meestal rijker dan geïsoleerde activiteiten omdat kinderen tijdens één activiteit grote stappen kunnen zetten op verschillende vlakken.

Tijdens activiteiten rond programmeren moeten kinderen zeer vaak samenwerken. Daarom zijn deze activiteiten zeer rijk op socio-emotioneel vlak. De kleuters leren niet enkel hoe ze moeten omgaan met elkaar, maar ook dat ze elkaar kunnen helpen om tot een gemeenschappelijk doel te komen en dat ze samen sterker zijn. In een maatschappij waar individualisering zeer populair wordt, is het belangrijk om de kleuters te leren dat het in groep ook fijn kan zijn.

Om samen te kunnen werken moeten kinderen uiteraard ook met elkaar communiceren. Dit zorgt ervoor dat kleuters hun taal kunnen ontwikkelen, via niet-talige activiteiten. Voor



introverte kinderen, vormen deze activiteiten ook een kans om in een kleine (veilige) groep aan het woord te komen.

Beweging kan ook zeker aan bod komen. De kleuters kunnen elkaar begeleiden in een doolhof door de juiste instructies te geven. Hierbij komt ook weer taal aan bod.

## Algoritmisch denken

Hierboven werd het proces om tot een geprogrammeerd toestel/voorwerpen te komen, met eenvoudige woorden geschetst. Hierbij werd een belangrijke stap overgeslagen omdat deze eerste stap een apart hoofdstuk verdient. De stap van het algoritmisch denken vormt de basis van het programma. Aan deze stap ontsnapt geen enkele programmeur.

In het dagelijks leven komt iedereen kleine problemen tegen. Vaak zijn dit banale problemen waarvoor we al reeds een oplossing hebben of waarover we niet lang moeten nadenken. Hieronder wordt de uitleg even ondersteund met behulp van een voorbeeld.

Persoon A heeft een nieuwe telefoon en zou graag een bericht sturen met een hartje op het einde. De eerste mogelijkheid is dat persoon A al weet hoe dit moet. Persoon A heeft in het verleden al eens een hartje op het scherm laten verschijnen en weet dus hoe het probleem opgelost wordt.

Een tweede mogelijkheid kan zijn dat persoon A niet lang moet nadenken om het probleem op te lossen. Dit komt omdat persoon B goed weet hoe een hartje op zijn scherm kan laten verschijnen. Persoon A probeert dit dan zelf uit en het probleem is opgelost.

Het kan ook zijn dat niemand een gelijkaardige telefoon heeft en dus niemand weet wat er gedaan moet worden om een hartje op het scherm te laten verschijnen. Persoon A moet dan gaan onderzoeken hoe het probleem opgelost kan worden. Dit is dan meteen de derde mogelijkheid en met andere woorden een algoritme.

De weg om van een probleemstelling tot een oplossing te komen, wordt een algoritme genoemd. (Algoritmisch denken, 2016)

Een programmeur is een zeer creatieve persoon. Om te programmeren moeten de ontwikkelaars constant probleemoplossend denken. Ze stellen een doel op en onderzoeken hoe ze hun doel kunnen bereiken.

Programmeurs onderzoeken bijvoorbeeld hoe ze een robot kunnen laten stappen, of hoe ze een personage in een spelletje kunnen laten springen.

Elk computerprogramma is eigenlijk één groot algoritme, maar dan omgezet in een programmeertaal. De programmeur zet zijn ideeën, oplossing om in instructies om zo een doel te bereiken.

Wanneer we een programma willen schrijven, zullen we dus steeds eerst het algoritme uitwerken. We willen namelijk niet in het wilde weg beginnen programmeren. Er zijn verschillende stappen om tot een algoritme en vervolgens het programma te komen.

We starten natuurlijk bij de probleemstelling: wat is het probleem? Het is ook belangrijk om tijdens deze stap na te gaan wat het resultaat moet zijn.

Vervolgens moet het probleem geanalyseerd worden. Het is belangrijk om te weten of er hindernissen zijn om tot het doel te komen (bijvoorbeeld: de knopjes van een telefoon werken niet meer, het is dus onmogelijk om het hartje te laten verschijnen). Het is ook zeer belangrijk en nuttig om na te gaan of er al een oplossing bestaat om het probleem op te lossen.

Daarna begint de stap van het algoritmisch schrijven. Een hypothese wordt ontwikkeld en getoetst. Vaak wordt het probleem ook in deeltjes opgesplitst om deze stap te vereenvoudigen.

Het algoritme, wordt dan eindelijk omgezet in een code.

Hierop volgt de testfase. Het programma wordt op verschillende manieren, door verschillende gebruikers uitgetest. Daarna kan het programma aangepast worden.

Wanneer het programma, het probleem perfect oplost en dus het doel bereikt wordt, begint de stap van het documenteren. Een code is zeer ingewikkeld, daarom is het belangrijk om commentaar bij de code toe te voegen. Hierdoor kan iemand anders of de ontwikkelaar zelf, de code in de toekomst opnieuw gebruiken. (Algoritmisch denken, 2016)

## Computationeel denken

De bovenstaande voordelen van het programmeren met kleuters zijn slechts, niet te onderschatten, nevendoelen. Deze doelen worden soms onbewust bereikt dankzij de activiteiten rond programmatie. Het is ook mogelijk om de bovenstaande doelen bewust te integreren in een programmatieactiviteit.

Het computationeel denken van de kleuters ontwikkelen, is het voornaamste doel van programmatieactiviteiten. Kleuters leren via allerlei activiteiten, problemen op te lossen. (Computational, 2015)

Hierbij is het heel belangrijk om een onderscheid te maken tussen computationeel denken en probleemoplossend denken. Het probleemoplossend denken, is een populaire techniek in het onderwijs. Sinds een aantal jaren wordt steeds meer aandacht geschonken aan deze techniek. Zowel kleuteronderwijzer(essen)s als hogescholen geven hieraan aandacht.

Het is de bedoeling dat jonge kinderen ervaring opdoen door geconfronteerd te worden met problemen op hun niveau. De kleuters worden gestimuleerd om zelf opzoek te gaan naar oplossingen op hun problemen door verschillende hypotheses uit te proberen. (Computational, 2015)

De leid(st)er biedt divers materiaal aan en geeft de juiste impulsen om de kleuters zo goed mogelijk te ondersteunen in dit proces.

Voor computationeel denken geldt exact hetzelfde. De leid(st)er heeft steeds een zeer belangrijke, ondersteunende rol.

Het verschil zit hem in het aangeboden materiaal en het denkproces. Computationeel denken is het procesmatig oplossen van problemen via computertechnologie of met andere woorden via programmatie. Bij dit proces komen kinderen in contact met ICT-technieken en gereedschappen. (Computational, 2015)

Een pop laten bewegen, bijvoorbeeld, kan zowel via probleemoplossend denken als computationeel denken. Bij probleemoplossend denken gaan de kleuters via een draadmechanisme zelf de pop laten bewegen. Bij computationeel denken gaan de kinderen de pop op de juiste manier programmeren en zo via computertechnologie de pop laten bewegen. (Computational, 2015)

### Waarom computationeel denken?

Veel van de huidige maatschappelijke en wetenschappelijke vraagstukken zijn zo complex dat het onmogelijk is deze vraagstukken op te lossen zonder de hulp van computertechnologie. Grafieken, formulieren en vele andere dagelijkse zaken worden via computertechnologie geregeld.

Een kleuter in de 21<sup>ste</sup> eeuw kan niet ontsnappen aan deze realiteit. Technologie is aanwezig in de huidige maatschappij, ook in de wereld van de jonge kinderen. (Computational, 2015)

Digitale geletterdheid in het onderwijs is zeer belangrijk en kinderen moeten er zo vroeg mogelijk mee in contact komen. Het is niet genoeg dat kinderen passief omgaan met deze technologieën zonder dat ze de achterliggende processen ervan, min of meer begrijpen.

Het is niet de bedoeling om van elk kind een toekomstige programmeur te maken, maar wel om elk kind actief te laten consumeren.

De huidige kleuters zijn de volwassenen van morgen en zullen in de toekomst in contact komen met nieuwe technologieën. Het is dus niet zinvol om ze enkel de computerprogramma's van nu te leren. Kinderen moeten de vaardigheid om te kunnen omgaan met elk digitaal middel, zo vroeg mogelijk ontwikkelen. Op deze manier creëren we een generatie van succesvolle techniekgebruikers. (Maas, 2016)

Door te leren programmeren, begrijpen kinderen hun speelgoed ook veel beter. De huidige speelgoedindustrie produceert zeer veel geprogrammeerd speelgoed. Kinderen die weten hoe hun speelgoed in elkaar zit komen niet per se tot beter spel, maar het is hun recht om hun speelgoed te begrijpen. Spel is primordiaal voor de ontwikkeling van een kind en hangt in de 21<sup>ste</sup> eeuw samen met digitale geletterdheid. (Maas, 2016)

### Computationeel denken in het dagelijks leven

Daarnaast is het belangrijk om de nadruk te leggen op het feit dat kinderen computationeel denken zeer goed kunnen gebruiken in hun dagelijks leven. Via de programmatieactiviteiten ontwikkelen ze zeer belangrijke vaardigheden die ze dagelijks kunnen toepassen.

Om te programmeren is het belangrijk om stapsgewijs aan de slag te gaan. Leren plannen is iets heel belangrijks in het kleuteronderwijs. Kinderen die zonder structuur aan de slag gaan en die niet gestimuleerd worden om te plannen, komen later in de problemen terecht.

In het secundaire onderwijs, wordt de hoeveelheid leerstof en taken enorm. In het hoger onderwijs gaat het natuurlijk nog een stapje verder. Studenten die niet kunnen plannen hebben het moeilijker om alles op tijd af te krijgen. Het leren plannen op jonge leeftijd heeft dus zeer positieve effecten op de rest van de schoolcarrière.

Ook buiten de schoolmuren is plannen zeer belangrijk. Om een juist evenwicht te creëren tussen gezin, werk en vrijetijd, is een goede structuur nodig.

Tijdens een programmatieactiviteit moet er altijd een plan gemaakt worden. Een soort hypothese die daarna uitgeprobeerd wordt. Hierbij wordt stapsgewijs gewerkt door een algemeen probleem te verdelen in kleinere problemen. Een probleem kan, bijvoorbeeld, zijn dat kinderen een robot willen laten stappen. Hiervoor moeten ze ook weten hoelang de robot moet kunnen stappen, op welk tempo, in welke richting.... Dit zijn dan allemaal kleinere problemen waarop een antwoord gevonden moet worden om het algemeen probleem efficiënter op te lossen.

Daarnaast is het belangrijk om te beseffen dat iedereen, ook kinderen, constant keuzes moet maken. Een maaltijd kiezen, een kledingstuk kiezen, een parfum kiezen zijn allemaal voorbeelden van de keuzes die we dagelijks moeten maken.

Kiezen is voor jonge kinderen vaak moeilijk en dat is iets dat ze moeten leren. Wanneer kinderen programmeren, zijn er vaak verschillende mogelijkheden. Ze moeten deze mogelijkheden met elkaar vergelijken om zo de beste optie te kiezen.

De juiste keuze kunnen maken is een vaardigheid die kleuters via programmatie kunnen ontwikkelen en die ze in hun dagelijks leven zeker kunnen gebruiken.

## **4. Hoe populair is programmeren momenteel in het onderwijs?**

Programmeren is sinds een aantal jaren enorm populair geworden. Sinds kort wordt ook in het onderwijs aandacht geschonken aan digitalisering, media en programmatie. Hieronder wordt een enquête besproken waaruit duidelijk blijkt dat deze nieuwe trend nog in zijn kinderschoenen staat, maar toch al goed op weg is.

In totaal hebben ééneveertig leerkrachten uit heel Vlaanderen de enquête ingevuld. De leerkrachten uit Brussel en Antwerpen waren het meest reactief. De enquête bestond uit vijf vragen rond programmeren met en door kinderen.

Achttien en zeventien van de respondenten onderwijzen respectievelijk, in een derde en tweede kleuterklas. Veertien respondenten onderwijzen in een onthaalklas of eerst kleuterklas. Leerkrachten bij de oudste kleuters voelde zich dus blijkbaar meer aangesproken. Dit spreekt ook voor zich, omdat echte activiteiten rond programmeren vaak bestemd zijn voor de oudere kleuters.

Op de derde en belangrijkste vraag van de enquête namelijk: "Hebt u al gehoord van programmeren met en door kleuters?", waren de antwoorden bijna gelijk verdeelt. Dit wordt zeer duidelijk weergegeven op het schijfdiagram. Eenenvijftig procent van de leerkrachten had al ooit gehoord van programmeren met en door kinderen. Negenveertig procent van de leerkrachten heeft hier nog niet van gehoord.

Hieruit kan geconcludeerd worden dat quasi één op twee leerkrachten al gehoord heeft van programmatie op kindermaat. Dit is een behoorlijk mooi resultaat als men weet dat het onderwerp redelijk nieuw is in het onderwijs. Uiteraard is het belangrijk om in het achterhoofd te houden dat het hier over een kleinschalig onderzoek gaat.

Bij de speelleermaterialen, was de Bee-Bot de grote winnaar met vijfentwintig respondent die al van deze programmeerbare robot gehoord hadden. Dat is dus opnieuw één op twee respondent. Hieruit is dus duidelijk gebleken dat de Bee-Bot nog altijd de populairste robot is in Vlaanderen.

Scratch is op de tweede plaats geëindigd, zeven op de eenenveertig respondenten hadden al van Scratch gehoord. De nieuwe Jack kwam op de laatste plaat met slechts drie respondent.

Deze resultaten waren eigenlijk voorspelbaar, deze drie speelleermaterialen worden in de komende hoofdstukken besproken en vergeleken. Het succes van de Bee-bot wordt daar ook verklaard.

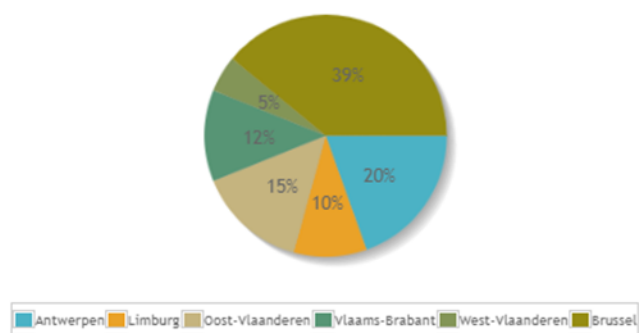
Het meest positieve uit deze enquête is zeker en vast het feit dat eenenvijftig procent van de leerkrachten open staat om een activiteit rond programmatie uit te proberen in de kleuterklas. Dit toont enerzijds aan dat programmeren op kindermaat de leerkrachten zeker interesseert. En anderzijds blijkt hieruit dat de Vlaamse leerkrachten openstaan voor innovatie. Innovatie is een belangrijk aspect van het onderwijs.

In het algemeen zijn de resultaten van deze enquête geruststellende en stralen hoop uit. Uit de resultaten kan geconcludeerd worden dat de leerkrachten open staan voor deze gloednieuwe trend en dat ongeveer één op twee leerkrachten er eigenlijk al mee weg is.

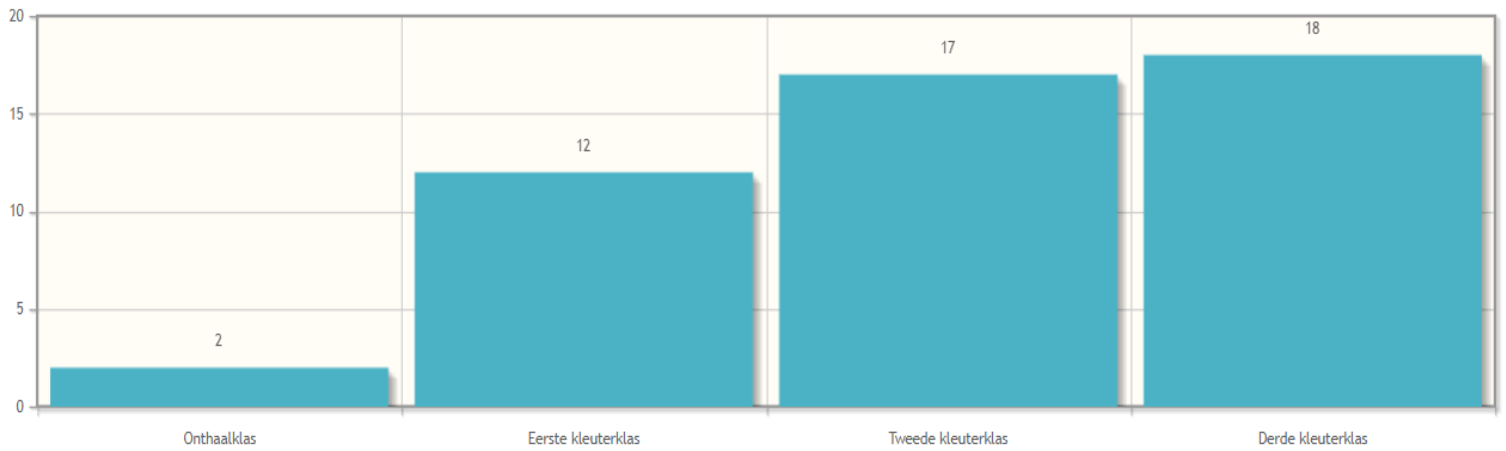
## Resultaten van de enquête

Zoals hierboven vermeld werd dus een enquête opgesteld om na te gaan hoe populair programmeren met en door kinderen momenteel is in Vlaanderen. Hieronder worden de resultaten van de enquête, per vraag, weergegeven in duidelijke grafieken.

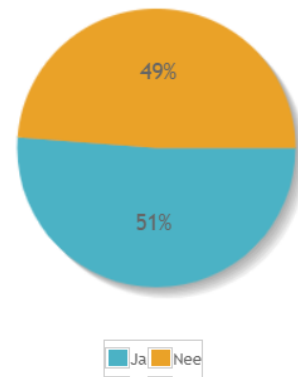
Statistieken voor vraag 1 : In welke provincie onderwijst u?



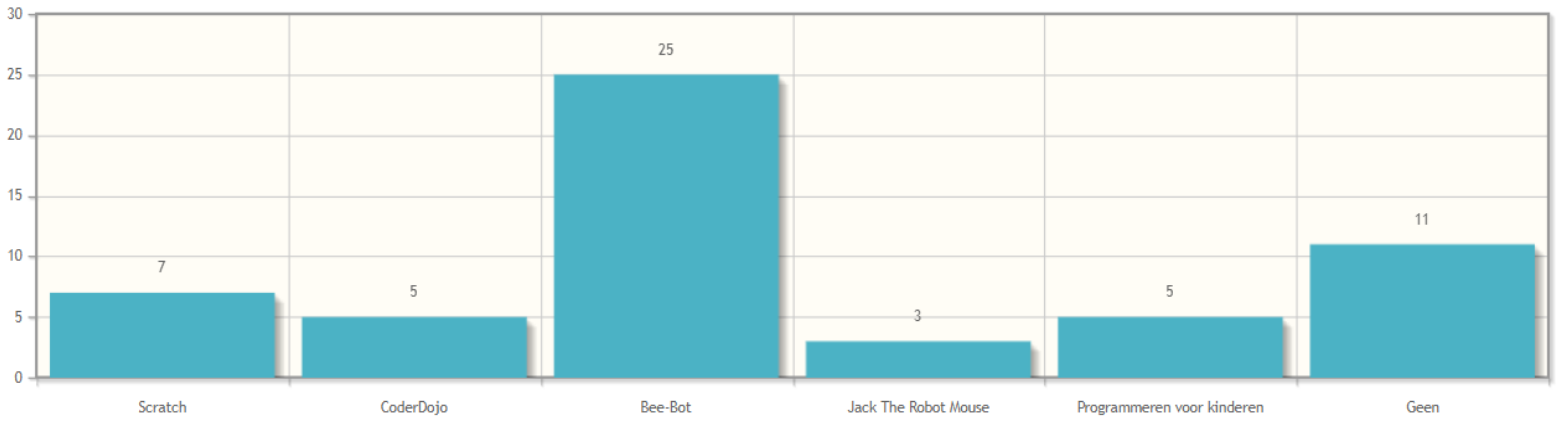
**Statistieken voor vraag 2 : In welke kleuterklas onderwijst u?**



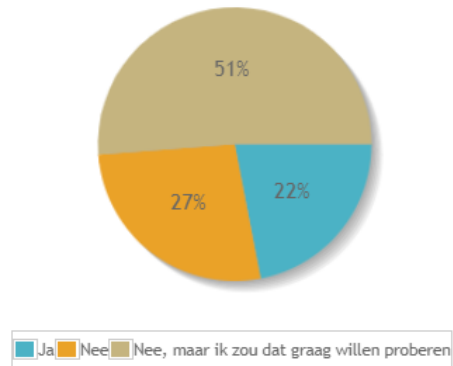
**Statistieken voor vraag 3 : Hebt u al gehoord van programmeren met en door kleuters?**



Statistieken voor vraag 4 : Hebt u al gehoord van één van onderstaande methode? Kruis aan.



Statistieken voor vraag 5 : Hebt u zelf al geprogrammeerd met kleuters, of activiteiten in verband met programmeren gedaan in uw kleuterklas?





## 5. Programmeren op kleuterniveau

In de afgelopen jaren is de belangstelling voor programmeren of coderen enorm toegenomen. Er worden programmeerclubs opgericht om beginners te laten kennis maken met codes. Veel volwassenen keren terug naar de schoolbanken om programmeervaardigheden aan te leren.

De populariteit is te danken aan de digitalisering van onze hedendaagse maatschappij, steeds meer voorwerpen worden geprogrammeerd.

Daarnaast spelen de hoge lonen van de programmeurs ook veel mensen aan. In een wereld waarin iedereen het steeds moeilijker heeft gaan mensen opzoek naar een beroep waarmee ze meer kunnen verdienen.

Maar miljoenen mensen ontdekken ook thuis hoe uitdagend en moeilijk programmeren is en deze uitdaging trekt de aandacht.

Zo zien we ook dat scholen over de hele wereld programmeren aan de leerstof toevoegen. Sinds een aantal jaar ontdekken ook lagere scholen en kleuterscholen de positieve effecten van coderen.

Dankzij nieuwe websites en programma's kunnen heel jonge kinderen aan de slag met programmeren. Meer en meer scholen en begeleiders van jonge kinderen maken gebruik van dit nieuw materiaal. Hieronder worden een aantal mogelijkheden om aan de slag te gaan met coderen met jonge kinderen, besproken.

### Bestaande robots en handleidingen

#### De Bee-Bot

De Bee-bot is de populairste programmeerbare robot op de markt. Zijn aantrekkelijk vorm en felle kleur spreekt zowel volwassenen als kinderen meteen aan. Deze robot was ook één van de eerste programmeerbare robots voor kinderen en heeft het publiek kunnen charmeren.

De Bee-bot is een kleine robot in de vorm van een bij. Met de Bee-bot kunnen jonge kinderen leren programmeren op hun niveau. Op zijn rug heeft de Bee-bot een aantal knopjes waarmee de kinderen instructies kunnen geven.

De kleuters moeten de robot tot een bepaald punt brengen, hiervoor moeten ze de robot programmeren. Via de knopjes kunnen de kleuters de Bee-bot de juiste aanwijzingen geven. De robot onthoudt tot veertig stappen achterelkaar. Om opnieuw te beginnen, kunnen de kleuters



*Figuur 1 De Bee-Bot*

op de reset knop drukken. Wanneer de Bee-bot zijn eindpunt bereikt heeft, maakt hij een leuk geluidje met een knipperend lichtje. (Marita, 2016)

Het succes van de Bee-bot is ook te danken aan al het materiaal die bestaat om samen met de Bee-bot te gebruiken. Er bestaan matten in quasi elk thema (boerderij, water, lente...). Dat is natuurlijk fantastisch voor een kleuterklas omdat het mogelijk wordt om de Bee-bot op deze matten te laten bewegen en activiteit zonder veel moeite te verrijken in het thema. Er bestaan ook matten rond cijfers, letters, kleuren en nog zoveel meer.

Daarnaast zijn er ook doolhoven en routekaarten verkrijgbaar op de markt. Deze zijn ook enorm populair omdat ze ervoor zorgen dat de opdrachten steeds moeilijker worden.

Toch heeft de Bee-bot een groot nadeel. Zijn prijs natuurlijk. De robot kost tussen honderdtwintig en vierhonderd euro. Dat is natuurlijk een belangrijk budget voor scholen, vooral omdat de robot niet zolang meegaat. (Bee-Bot, 2016)

### Jack the robot mouse:

De grootste concurrent van de Bee-bot is zonder twijfel Jack de muis. Jack heeft op vak van uiterlijk niets te verwijten aan de Bee-bot. Ook hij maakt de kleuters zeer snel enthousiast en trekt meteen de aandacht.



*Figuur 2 Jack The Robot Mouse*

Op vlak van mechanisme kunnen Jack en de Bee-bot ook exact hetzelfde.

Jack kost twee keer minder als de Bee-bot. Dit is natuurlijk de rede waarom Jack zo snel populair is geworden. Deze robot kost ongeveer dertig euro en een heel pedagogisch pakket met, doolhof en routekaarten kost minder als vijftig euro. Een volledig Bee-bot pakket kost vierhonderd euro, dat is natuurlijk niet vergelijkbaar. (Mouse, 2013)

De vraag is nu natuurlijk waarom Jack slecht een concurrent van de de Bee-bot is en deze laatste nog niet onttroond heeft. Hiervoor bestaan twee verklaringen.

Zoals hierboven vermeld, was de Bee-bot de eerste programmeerbare robot voor kinderen op de markt. Er werd een paar jaar geleden heel veel over de Bee-bot gesproken. Zo hebben vele leerkrachten de Bee-bot leren kennen.

Over de andere robots, na de Bee-bot, werd veel minder gesproken omdat ze niet revolutionair waren. Omdat de Bee-bot de eerste op de markt was en omdat er een paar jaren geleden zoveel over deze robot gesproken werd, is hij blijven hangen in het geheugen van de leerkrachten. Als ze nu willen programmeren met hun kleuters, denken ze meteen aan de Bee-bot.

Het is een beetje vergelijkbaar met grote merken zoals Coca-Cola. Er bestaan nu een tiental identieke frisdranken voor een lagere prijs, maar toch blijven mensen Coca-Cola kopen. Dit is een bekend fenomeen in de commerciële wereld.

Een tweede belangrijk argument is het feit dat Jack, over weinig aangepast materiaal beschikt. Er bestaat wel een mat met vakjes, een aanpasbare doolhof en een aantal routekaarten, maar meer niet. Het is natuurlijk zeer gemakkelijk om zelf extra-materiaal aan te maken, maar dit vraagt tijd. (Mouse, 2013)

Voor de Bee-bot, bestaat er zeer divers materiaal. Er bestaat ook materiaal voor quasi elk thema. Voor scholen is dit wel een belangrijk voordeel omdat het makkelijker is om iets aan te kopen, dan zelf de tijd te nemen om het te maken.

## Scratch



Scratch is een zeer populaire website waarmee kinderen de beginselen van het programmeren kunnen leren door het maken van een project. Elk project is eigenlijk een spel die de kinderen zelf kunnen creëren. In het jargon van de toegepaste informatica wordt een spel vaak een project genoemd.

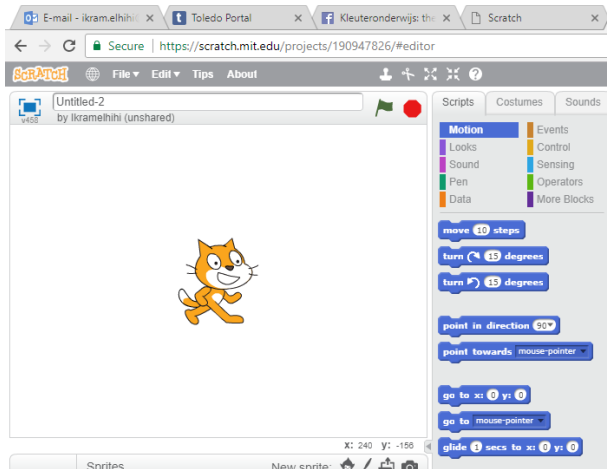
*Figuur 3 Scratch*

De kinderen kunnen een figuur kiezen en via de computer opdrachten geven om de figuren te laten bewegen, geluid laten maken en te reageren. Het toepassingsgebied is breed: er kunnen animaties gemaakt worden, games, verhalen, muziek, simulaties, etc. (Scratch, 2016)

Vooraleer de kinderen aan een eigen project kunnen starten, moeten ze een aantal tutorial volgen. Deze tutorial zijn zeer duidelijk en aangepast aan kinderen. De kinderen leren dan in elk tutorial iets nieuw.

Bij de eerste tutorial kunnen ze leren hoe ze figuren kunnen laten stappen, in tutorial van hogere niveaus kunnen ze zelfs al leren hoe ze een figuur kunnen laten spreken. De tutorials zijn niet verplicht, maar starten zonder tutorial is zeer ingewikkeld omdat er zoveel mogelijkheden zijn. De kinderen zullen niet weten waarmee ze moeten beginnen en hoe ze de tools op de juiste manier moeten gebruiken. (Website Scratch, 2017)

Scratch is een methode die bestemd is voor lagereschoolkinderen die al kunnen lezen. Om een eigen project te starten is het echt noodzakelijk om te kunnen lezen omdat de kinderen telkens een actie moeten aanvinken. Hieronder een afbeelding die aantoont hoe de kinderen een bepaalde actie kunnen aanvinken.



Figuur 4 Voorbeeld van acties bij Scratch

Dit betekent uiteraard niet dat Scratch geen inspiratiebron kan zijn om te programmeren met kleuters. Op de website zijn ook kaarten beschikbaar, die ook gebruikt kunnen worden bij robots zoals Jack en de Bee-Bot. Sommige kaarten zijn ook een mooie inspiratiebron voor een drama activiteit voor kleuters. Hier wordt later nog dieper ingegaan.

Daarnaast zijn er nog een aantal handleidingen voor leerkrachten te vinden. Deze handleiding bevatten ook heel wat leuke inspirerende activiteiten die aangepast kunnen worden aan het niveau van kleuters.

De meeste handleidingen en kaarten zijn ook gratis beschikbaar, het is dus zeker de moeite waard om er een kijkje naar te nemen.

## Programmeren voor kinderen

Programmeren voor kinderen is een zeer populaire handleiding. In de handleiding kunnen kinderen en volwassenen kennismaken met de wereld van de programmatie.

In de handleiding worden de bekendste programmeertalen kort overlopen. Ook de meest voorkomende coderingswoorden zoals algoritme, bug en code worden mooi uitgelegd. Op deze manier kan de lezer een stevig basis creëren om daarna zelf aan de slag te gaan.

Programmeren voor kinderen werd geschreven om een groot publiek te laten kennismaken met de creatieve mogelijkheden van computers. Via de handleiding kunnen de lezers aan de slag gaan met kunst, muziek, dans en nog veel meer. (Woodcock, Vorderman, 2017)

Om hun doel te bereiken hebben de schrijvers gekozen voor het programma: Scratch. In het vorig hoofdstuk werd scratch al kort besproken. Via Scratch kunnen de gebruikers programma's en projecten maken door al bestaande instructieblokken in elkaar te plaatsen. De gebruiker hoeft dus niet meer zelf een code te schrijven, maar maakt gebruik van bestaande codestukjes.

Dit is een revolutionaire vooruitgang in de wereld van computers. Dankzij dergelijke programma's kan een steeds groter en jonger publiek kennismaken met programmeren.



Programmeren voor kinderen is eigenlijk een handleiding waarin alle mogelijkheden van Scratch terug te vinden zijn. Dit kan heel handig zijn voor leerkrachten die aan de slag willen met Scratch in hun klas.

Voor kinderen zelf is de handleiding niet ideaal. Voor leerkrachten die opzoek zijn naar een inspirerend werk, vormt de handleiding een mooie inspiratiebron. (Woodcock, Vorderman, 2017)

*Figuur 5 Programmeren voor kinderen*

### De Schrijfster

De handleiding programmeren voor kinderen werd geschreven door Carol Vorderman. Deze dame is één van de populairste Britse tv-presentatrice. Daarnaast is ze ook een zeer bekende wiskundige.

Zij verdient in dit werk wel een apart hoofdstuk omdat zij eigenlijk de nieuwe trend om kinderen te leren programmeren heeft gelanceerd.

Ze pleit er al jaren voor dat elk kind de kans moet krijgen de waardevolle wereld van het coderen te leren kennen.

Via haar adviserende rol aan de voormalige Britse premier David Cameron, bleek ook dat ze op een spannende en voor iedereen te begrijpen manier, wiskunde, wetenschappen en vooral technologie wou over brengen op het grote publiek. (Woodcock, Vorderman, 2017)

## **6. Eigen activiteiten rond programmeren :**

In bibliotheken en op het internet zijn heel wat inspirerende programma's en handleidingen te vinden om te werken rond programmeren met en door kinderen. Een aantal van deze inspiratiebronnen werden in de vorige hoofdstukken besproken.

De handleidingen en programma's zijn meestal zeer gemakkelijk te vinden en gratis beschikbaar. Dit is een zeer groot voordeel en dit zou heel wat leerkrachten moeten aanmoedigen om te programmeren met kleuters.

Maar uit de enquête die hierboven werd besproken, blijkt dat programmeren met en door kinderen nog niet zo populair is in Vlaanderen. Het onderwerp is in Vlaanderen wel bekend, maar weinig leerkrachten durven het zelf uit te proberen in de klas.

Het is belangrijk om de leerkrachten in Vlaanderen te informeren over de verschillende handleiding, programma's, websites en dergelijke die een houvast kunnen bieden om te beginnen aan programmatie in de klas.

De leerkrachten in spe van de opleiding kleuteronderwijs aan de Odisee hogeschool, zijn de voornaamste doelgroep van dit werk. De studenten staan nog open voor nieuwe lesonderwerpen en nieuwe technieken om de hedendaagse technologieën in de klas te brengen. De toekomstige leerkrachten zijn een zeer goede doelgroep omdat zij vernieuwing kunnen brengen in het onderwijs. Om een nieuwe techniek populair te maken, moeten deze toekomstige leerkrachten geraakt worden.

Om de doelgroep zo goed mogelijk te begeleiden en een stevig houvast te bieden, hoort bij dit werk ook een handleiding. De handleiding bestaat uit verschillende activiteitenfiches rond programmeren met kleuters.

De activiteitenfiches zijn uitgewerkt volgens de regels van de hogeschool. Bij elke activiteit zijn een aantal Zill-doelen en ontwikkelingsdoelen terug te vinden. Ook wordt er bij elke activiteit aandacht besteed aan woordenschat, probleemoplossend denken, planmatig werken en vakoverschrijdend werken.

Deze activiteitenfiches hebben als bedoeling om de studenten aan te moedigen om te werken rond programmeren in de klas. De studenten hebben dan een aangepaste en volledige inspiratiebron ter beschikking.

De handleiding bestaat uit activiteitenfiches met of zonder robot. Alle activiteiten die uitgewerkt zijn voor robots, kunnen uitgevoerd worden met eender welke robot. Dit is van belang, omdat de leerkrachten dan volgens hun eigen budget een robot kunnen kiezen.

De activiteiten zonder robot, zijn ook zeer makkelijk uit te voeren in de praktijk en vragen weinig of makkelijk zelf te maken materiaal.

## **7. Het thema robots**

In dit stukje wil ik even mijn persoonlijke ervaring meedelen. Tijdens mijn uitgroei stage werk ik het thema: "Robots" uit. Dit thema past perfect bij programmeren met en door kleuters. Robots spreken kinderen aan en fascineren hen.

Ik werk een hele week het thema uit. Uiteraard is het mogelijk om programmeren in elk thema te integreren, in de fiches van de handleiding staan dan ook heel wat voorbeelden om vakoverschrijdend aan de slag te gaan.

Ik koos voor het thema robots in plaats van het thema programmeren, omdat bij het thema robots ook andere activiteiten aan bod kunnen komen. Het is belangrijk om niet enkel programmatie activiteiten aan te bieden, omdat sommige kinderen misschien niet geïnteresseerd zullen zijn in deze activiteiten. Dankzij het thema robots kan ik dus dagelijks activiteiten integreren in mijn aanbod die niet echt te maken hebben met programmatie, maar toch bij het thema robots horen. Voor kinderen die minder geïnteresseerd zijn in het thema programmeren, creëren deze activiteiten een mooie kans om toch betrokken te worden bij het thema.

### **De kleuters**

Het thema robots werk ik uit in een graadklas met vier en vijfjarige kleuters. De kleuters hebben enorm veel interesse in robots, ze vinden robots geweldig. Op hun school heeft een andere klas al gewerkt rond het thema robots, sindsdien zijn de kleuters intrinsiek gemotiveerd om aan de slag te gaan met het thema robots.

### **Samenwerking met een software ingenieur**

Ik heb de kans om tijdens het thema robots, samen te werken met een echte software ingenieur. Dit is een enorm voordeel omdat we de kennis van een specialist kunnen gebruiken. Onder het motto "de leraar als partner van externen" is deze kennis altijd welkom.

Het voordeel is dat we dankzij Abdel, de software ingenieur, echte onderdelen van een robot gaan kunnen waarnemen. Deze zijn dure en moeilijk te vinden voorwerpen die ik zonder hem nooit naar de klas had kunnen brengen.

Uiteraard kan Abdel ons ook helpen en zijn ervaring delen in verband met het coderen. Opnieuw gaan de kinderen dankzij hem, in contact komen met echte codes.

Abdel wordt echt de rode draad van onze week. Elke dag maakt hij een filmpje waarin hij de kleuters opdrachten geeft die ze doorheen de dag kunnen uitvoeren. Hij zal hen ook elke dag een stukje van zijn wereld, zijn werk tonen en hierdoor worden de kleuters echt ingedompeld in de wereld van computers en technologieën.

## Opbouw van de week

Omdat de kleuters nog nooit gewerkt hebben rond programmatie, is het zeer belangrijk om de week stapsgewijs op te bouwen. De kleuters moeten in stapjes in contact gebracht worden met programmatie.

Het heeft absoluut geen zin om de kinderen meteen zelfstandig aan de slag te laten gaan met een programmeerbare robot. Dit zou veel te moeilijk zijn en hierdoor zouden de kinderen snel afhaken.

In het begin zullen de kleuters vooral kennismaken met de ingenieur. Ze leren zijn beroep kennen, ontdekken waar en hoe robots gemaakt worden en dergelijke. De kleuters gaan dus eerst veel impressies op doen, we werken van impressie naar expressie.

De programmeerbare robot, zal ik inleiden rond de tweede dag. We starten klassikaal zodat iedereen de kans heeft om op hetzelfde moment kennis te maken met de robot.

In het begin zullen alle activiteiten in verband met de robot plaatsvinden in de kring met de hele groep. Dit is belangrijk omdat de kleuters onder begeleiding eerst moeten leren werken met de robot, ze leren van elkaars fouten en zien allemaal hoe het moet.

Later in de week, zullen ze in kleinere groepjes aan de slag mogen met de robot. De robot zal na het thema robots uiteraard ook niet verdwijnen. De kleuters mogen ook tijdens andere thema's aan de slag met de programmeerbare robots.

Zoals hierboven vermeld zal er ook een rode draad ontstaan dankzij de samenwerking met de ingenieur. Hij zal hen elke dag via een filmpje, opdrachten geven die ze doorheen de dag mogen uitvoeren. Als ze de opdrachten allemaal goed uitvoeren, krijgen ze op het einde van de dag een prent van een robot. Ze moeten in totaal twintig robots verzamelen, dan pas kan de ingenieur naar de klas komen om hun eigen robot te laten zien.

Op deze manier creëren we ook een samenhorigheidsgevoel bij de kleuters. Ze moeten allemaal goed samenwerken om alle robots te verzamelen en ervoor te zorgen dat Abdel naar de klas komt.

## Voorbeeld weekschema en ideeën

Hieronder plaats ik een voorbeeld van een weekschema om het thema "robots" uit te werken. Uiteraard is dit slecht een voorbeeld en kan er nog veel meer gedaan worden. Het is zelfs mogelijk om hier twee weken rond te werken.



In het onderstaande weekschema, ligt de focus op programmeren. De meeste activiteiten gaan dus over programmeren en komen uit de handleiding. Uiteraard zijn er ook andere activiteiten uit andere ontwikkelingsdomeinen.

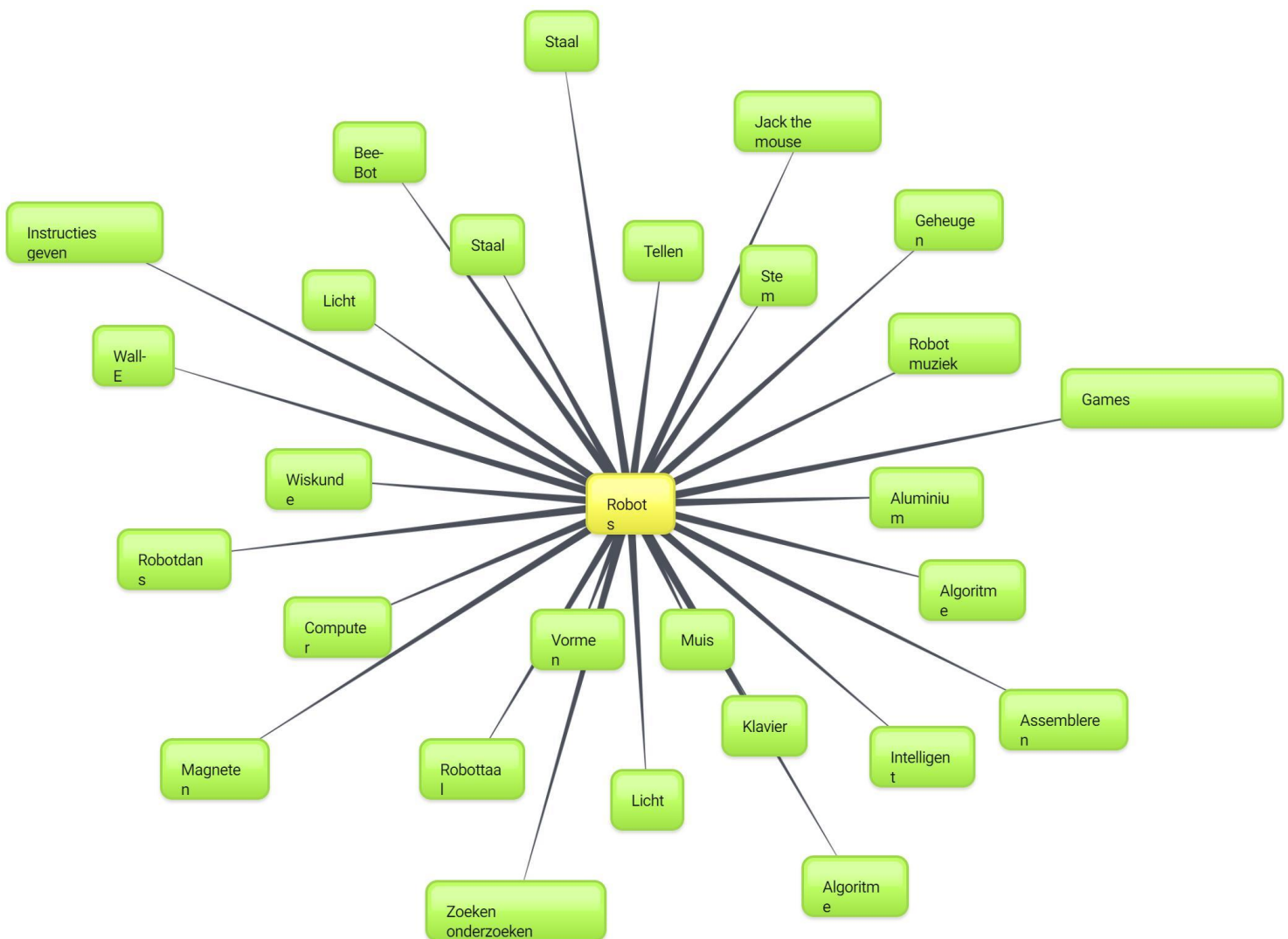
Bij het weekschema horen ook een brainstorm en een mindmap.

### Losse brainstorm

De losse brainstorm is een brainstorm waarin ik alle woorden noteerde die in mij opkwamen toen ik aan het woord robot dacht. Op deze manier heb ik al mijn ideeën neergeschreven.

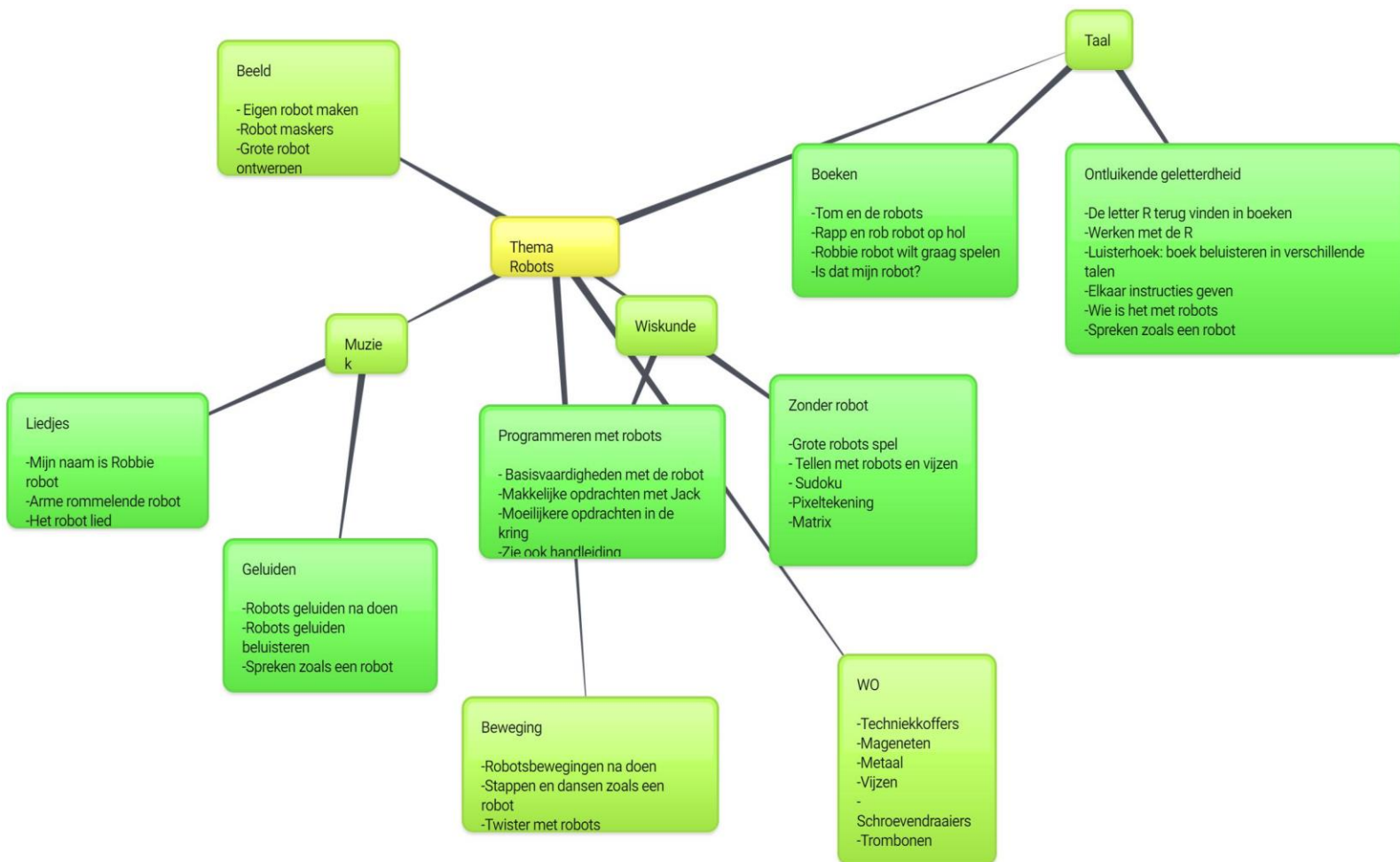
### Mindmap per ontwikkelingsdomein

Na het maken van de losse brainstorm, noteerde ik al mijn ideeën en concrete activiteiten in een mindmap.



Deze mindmap bestaat uit verschillende vakjes die de verschillende ontwikkelingsdomeinen illustreren.

Alle ideeën zijn natuurlijk slechts voorbeelden, er bestaan nog talloze andere ideeën rond het thema "robots".



created with [www.bubbl.us](http://www.bubbl.us)

## Voorbeeld weekschema

Hieronder plaats ik een voorbeeld van een weekschema rond het thema "Robots". Dit weekschema heb ik zelf gebruikt tijdens mijn uitgroei stage in een graadklas (vierjarige en vijfjarige). Zoals hierboven vermeld, ligt de focus in dit schema op het programmeren en de meeste activiteiten komen uit de handleiding.

Maandag 14 Mei	Dinsdag 15 Mei	Woensdag 16 Mei	Donderdag 17 Mei	Vrijdag 18 Mei	
HALO IK BEN JACK	Ik word een robot	Droom jij ook van een eigen robot?	Van links naar rechts, volg mijn instructies maar!	Robotland komt naar ons!	
<b>Onthaal</b>					
Introductie Jack: Wie zit er in de doos?	Groot Spel uitleggen	Eigen robot ontwerpen	Kan jij de robot naar de juiste prent brengen?	Workshop digitale wolven	
OOL: Basisvaardigheden met Jack (4) ZS Strijkparels met robotvormen (2) OOL: Tellen met robots en vijzen (2) ZS Memory met robots (3) ZS Boekenhoek (2) ZS Poppenhoek (2)	OOL Groot spel: ik ben een robot (4) ZS Boekenhoek (2) EB: Techniekkoffer met magneten (2) ZS Poppenhoek (4) OOL Makkelijke opdrachten met Jack : test-activiteit (2)	OOL Eigen robot maken met kosteloos materiaal (4) ZS Boekenhoek (2) OOL Kleurensudoku (2) ZS Poppenhoek (2) OOL Makkelijke opdrachten met Jackt (2) OOL Pixeltekening (2)	OOL Taal: Instructies geven aan elkaar adhv kunstwerken (4) ZS Boekenhoek (2) ZS: Luisterhoek zoek de fout (2) OOL Groot spel: ik ben een robot (4) ZS Poppenhoek (2) OOL Pixelteking (2)		
Speeltijd 9u50 tot 10u05 Fruit eten 10u05 tot 10u15					
Verhaal vertellen	Lied van de week aanleren Robbie Robot	Robotdans	Samen een koekje eten door instructies te geven/ volgen		Workshop digitale wolven
OOL: Basisvaardigheden met Jack (4) ZS Strijkparels met robotvormen (2) OOL: Tellen met robots en vijzen (2) ZS Memory met robots (3) ZS Boekenhoek (2) ZS Poppenhoek (2)	OOL Groot spel: ik ben een robot (4) ZS Boekenhoek (2) EB: Techniekkoffer met magneten (2) ZS Poppenhoek (4) OOL Makkelijke opdrachten met Jack : test-activiteit (2)	OOL Eigen robot maken met kosteloos materiaal (4) ZS Boekenhoek (2) OOL Kleurensudoku (2) ZS Poppenhoek (2) OOL Makkelijke opdrachten met Jackt (2) OOL Pixeltekening (2)	OOL Taal: Instructies geven aan elkaar adhv kunstwerken (4) ZS Boekenhoek (2) ZS: Luisterhoek zoek de fout (2) OOL Groot spel: ik ben een robot (4) ZS Poppenhoek (2) OOL Pixelteking (2)		
Naar de refter: 11u40 tot 12u			Naar de refter: 11u40 tot 12u		
Speeltijd 12u tot 13u			Speeltijd 12u tot 13u		
OOL: Basisvaardigheden met Jack (4) ZS Strijkparels met robotvormen (2) OOL: Tellen met robots en vijzen (2) ZS Memory met robots (3) ZS Boekenhoek (2) ZS Poppenhoek (2)	Turnen tovenaars		Turnen Toveraars	Vrij spel	
Speeltijd: 13u55 tot 14u10			Speeltijd: 13u55 tot 14u10		
OOL: Basisvaardigheden met Jack (4) ZS Strijkparels met robotvormen (2) OOL: Tellen met robots en vijzen (2) ZS Memory met robots (3) ZS Boekenhoek (2) ZS Poppenhoek (2)	Turnen draken		Turnen Draken		
Robot, Links, Rechts	Rebus Woord Robot	De R van Robot	Lied herhalen		

## 8. Aan de slag in de klas

Tijdens mijn uitgroeastage heb ik heel wat activiteiten kunnen uitproberen in de klas. Het thema robots werd een groot succes.

Programmeren was in het begin zeer moeilijk en de kleuters waren niet meteen mee. Ze waren wel allemaal onder de indruk van de robots en zeer gemotiveerd om mee te doen.

Hieronder een aantal sfeerbeelden.

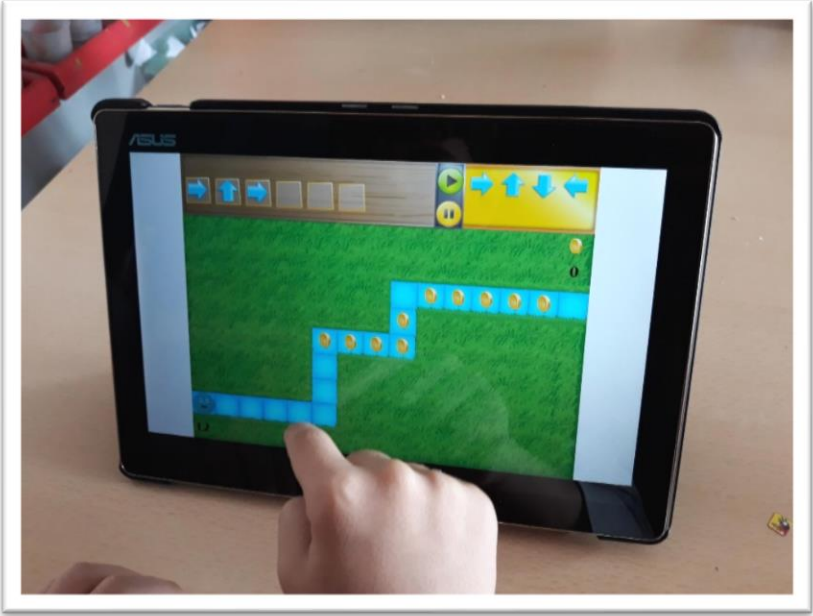
### Programmeren met Jack



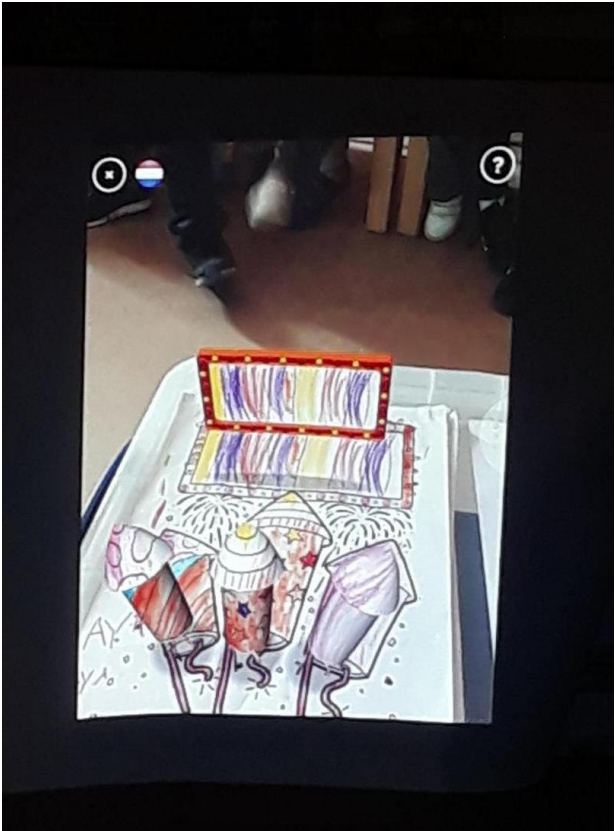
### Programmeren met de Bee-Bot



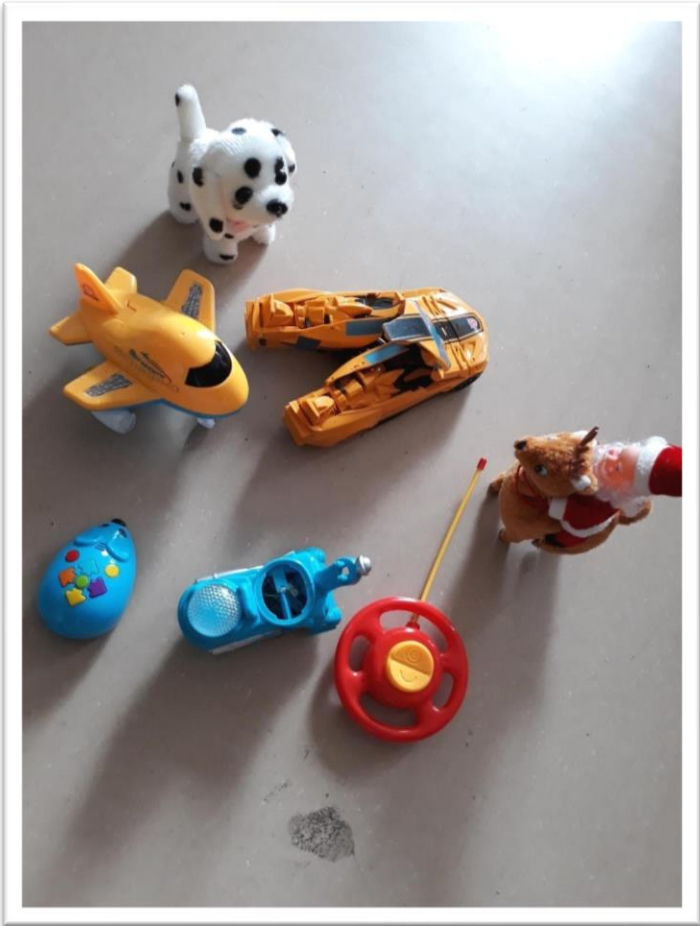
Programmeren op de tablet



3D-tekeningen maken



Eigen robot meebrengen naar de klas



## 9. Alles in een notendop

Programmeren is dus zeker en vast integreerbaar in een kleuterklas. Jongen kinderen vanaf vier jaar zijn in staat om aan de slag te gaan met eenvoudige algoritmes en codes.

Op de markt bestaan talloze handleiding en speelleermateriaal om aan de slag te gaan met programmatie in de kleuterklas.

De populaire robots zijn ook zeer interessant om me aan de slag te gaan in de kleuterklas. Hierboven werden de populairste robots vergeleken op vlak van kwaliteit en kostprijs.

Het is met andere woorden niet moeilijk om programmeren te integreren bij de oudste kleuters. Het is als leerkracht geen voorwaarde om er zelf veel over te weten of zelf te kunnen programmeren.

Programmeren kan geïntegreerd worden in quasi elk thema en er kan natuurlijk een hele week of meer gewerkt worden rond programmatie.

De voordelen van programmeren met en door kinderen zijn zeer divers. Naast het ontwikkelen van de creativiteit en het probleemoplossend vermogen, leren kinderen via programmatie ook tellen, woordenschat, zich situeren in de ruimte en nog veel meer.

Om te eindigen nog een citaat van Steve Jobs, dat heel dit werk mooi samenvat.

“Everybody in this country should learn how to program a computer... because it teaches you how to think”

Steve Jobs





## 10. Bronvermelding

- Boek, syllabus, brochure... met auteur(s)
- Duparcq, G. ( 2012). *Media in de kleuterklas*. Eindproef: Brussel
- Woodcokc, J. Vorderman,C. (2017). *Projects : programmeren voor kinderen*. Lannoo Uitgeverij 2017
- Digitale werken met auteur
- Maas, S. (2016). *Iedereen kan kinderen leren programmeren*. Geraadpleegd op 15 augustus 2017 via <https://www.klasse.be/39045/kinderen-leren-programmeren/>
- Marita. (2016). *Programmeren met kleuters met de Bee-bot*. Geraadpleegd op 22 augustus 2017 via <https://jufmarita.com/ict-en-tablets/programmeren-met-kleuters-met-de-bee-bot/>
- Vos,B. (2015). *15 PROGRAMMEERTALEN DIE JE MOET KENNEN IN 2015*. Geraadpleegd op 14 augustus 2017 via <http://www.youtech.nl/15-programmeertalen-die-je-moet-kennen-in-2015/>
- Digitale werken zonder auteur
- Algoritmisch denken. (2016). *Inleiding programmeren: Wat is algoritmisch denken?*. Geraadpleegd op 14 augustus 2017 via <https://informaticalessen.be/programmeren/wat-is-algoritmisch-denken/>
- Bee-Bot. (2016). *Bee-Bot*. Geraadpleegd op 16 augustus 2017 via <https://www.codekinderen.nl/leerling/programmeren/bee-bot/index.html>

- Computational. (2015). *Computational thinking*. Geraadpleegd op 15 augustus 2017 via <http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking>
- Mouse. (2013). *Code & Go™ Robot Mouse Activity Set*. Geraadpleegd op 16 augustus 2017 via <https://www.learningresources.com/product/learning+essentials--8482-+stem+robot+mouse+coding+activity+set.do>
- Programmeren. (2017). *Programmeren: computers*. Geraadpleegd op 24 augustus 2017 via [https://nl.wikipedia.org/wiki/Programmeren\\_\(computer\)](https://nl.wikipedia.org/wiki/Programmeren_(computer))
- Scratch. (2016). *Scratch*. Geraadpleegd op 26 november 2017 via <https://www.codekinderen.nl/leerling/programmeren/scratch/>
- Website Scratch. (2017). *Le monde de Scratch*. Geraadpleegd op 26 november 2017 via <https://scratch.mit.edu/>