HoGent

Faculteit Mens en Welzijn

DE INVLOED VAN EDUCATIEVE PROGRAMMEERTOOLS OP DE INTERESSE IN PROGRAMMEREN BIJ JONGEREN

Lise Vergauwen

Scriptie voorgedragen tot het bekomen van de graad van

Bachelor in het onderwijs: secundair onderwijs

Promotor:
Annick Van Daele

Instelling: Hogeschool Gent

Academiejaar 2016-2017

Derde examenperiode

Deze bachelorproef/scriptie is gemaakt door Lise Vergauwen, student aan de Hogeschool Gent, ter voltooiing van Bachelor in het onderwijs: secundair onderwijs. De standpunten die in deze bachelorproef/scriptie zijn verwoord, zijn louter het persoonlijke standpunt van de individuele auteur en reflecteren niet noodzakelijkerwijs de mening, het officiële standpunt of het beleid van de Hogeschool Gent.

Student: Lise Vergauwen
Promotor: Annick Van Daele
Academiejaar 2016 - 2017
Professionele Bachelor in het Onderwijs: Secundair onderwijs

HoGent
MENS
EN
WELZIJN

De invloed van educatieve programmeertools op de interesse in programmeren bij jongeren

Hogeschool Gent
Faculteit Mens & Welzijn
Opleiding Bachelor in het Onderwijs: Secundair onderwijs
Campus Ledeganck
K.L. Ledeganckstraat 8 | 9000 Gent
E. fmw@hogent.be | W. http://fmw.hogent.be

Student: Lise Vergauwen
Promotor: Annick Van Daele
Academiejaar 2016 - 2017
Professionele Bachelor in het Onderwijs: Secundair onderwijs

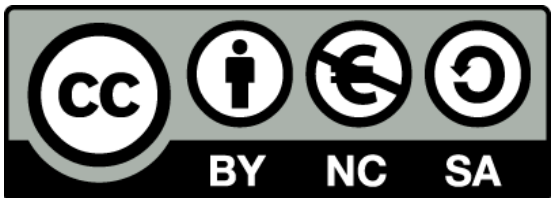
HoGent
MENS
EN
WELZIJN

De invloed van educatieve programmeertools op de interesse in programmeren bij jongeren

Hogeschool Gent
Faculteit Mens & Welzijn
Opleiding Bachelor in het Onderwijs: Secundair onderwijs
Campus Ledeganck
K.L. Ledeganckstraat 8 | 9000 Gent
E. fmw@hogent.be | W. http://fmw.hogent.be

Deze bachelorproef mag gebruikt worden indien voldaan wordt aan onderstaande Creative Commons licentie van het niveau:

'Naamsvermelding – Niet-commercieel – Gelijk Delen'.



Ook het logo van HoGent moet behouden blijven.

De volledige licentieovereenkomst kan geraadpleegd worden op:

http://creativecommons.org/licenses/by-nc-sa/2.0/be/legalcode.nl

Voorwoord

Beste lezer

Deze bachelorproef bevat een aantal methodes om programmeren aan te brengen bij kinderen en jonge adolescenten. Het tracht een antwoord te geven op de vraag welke methode de interesse het meest stimuleert, en welke tools hierbij het best aansluiten. Deze bachelorproef resulteerde onder andere ook in een website waarop u didactische tips vindt en enkele tools, gesorteerd per leeftijdsgroep.

In dit voorwoord past ook een welgemeende dank aan mijn promotor, Annick Van Daele. Ik wil haar bedanken voor alle hulp en inspiratie tijdens het maken van deze bachelorproef. Daarnaast wil ik graag mijn ouders, Greta Vercauteren en Werner Vergauwen, bedanken voor de steun en het nalezen van deze bachelorproef.

Ik hoop dat deze bachelorproef u zal kunnen inspireren, en dat u de invoer van programmeren in het onderwijs zal toejuichen.

Ik wens u veel leesplezier.
Lise Vergauwen

Inhoudstafel

Voorwoord **5**

Inleiding **8**

1 Literatuurstudie **13**

1.1 Waarom dit onderzoek? 13

1.2 Wat is programmeren? 16

1.2.1 Tekstueel versus visueel 16

1.2.2 Output van visueel programmeren 16

1.2.3 Motivatie 17

1.3 Wereldwijd 17

1.4 Leeftijden 18

1.5 Begeleiding 18

1.5.1 Feedback 18

1.5.2 Ontdekkend leren 18

1.5.3 Team 19

1.5.4 Project 19

1.6 Gebruikte tools 19

1.6.1 Scratch 19

1.6.2 Lego WeDo 21

1.6.3 Lego Mindstorms 22

1.6.4 Dwengo-bord 23

2 Onderzoeksopzet **24**

2.1 Progra-MEER 24

2.2 Leerkrachten 24

2.3 CoderDojo 24

3 Resultaten **25**

3.1 Progra-MEER 25

3.2 Leerkrachten 26

3.3 CoderDojo 31

4 Eindbesluit **36**

4.1 Hoofdonderzoeksvraag 36

4.2 Deelvragen 36

5 Reflectie **37**

Bibliografie **39**

6 Enquêtes via Google Forms **42**

6.1 Progra-MEER 43

6.2 Leerkrachten 48

6.3 CoderDojo 66

7 Website **76**

Inleiding

De meeste jongeren hebben vandaag de dag weinig tot geen interesse in programmeren. Terwijl er op de arbeidsmarkt steeds meer nood is aan computerspecialisten, daalt of stagneert de instroom van informaticastudenten aan de universiteiten. De Finse computerspecialist Linda Liukas is bang dat er een kloof zal ontstaan tussen gebruikers en makers, ofwel tussen hen die technologie louter gebruiken en zij die naast het gebruik ervan de technologie ook ontwikkelen (Het Nieuws van de Vooruitgang, 2016). Omdat we ons steeds meer met technologie omringen (Maas, 2015), is het misschien wel noodzakelijk om toch een basiskennis te hebben van hoe deze technologie werkt. “Wanneer je je een voorstelling kunt maken van wat mogelijk is, wordt het eenvoudiger om met die nieuwe technologie om te gaan” (Maas, 2015, p. 12).

Wanneer we de leerplannen van informatica bekijken (Vlaams Verbond van het Katholiek Secundair Onderwijs, 2011, p. 19; Onderwijs van de Vlaamse Gemeenschap, 2012, p. 15 – 16) merken we op dat er wordt ingezet op “algoritmisch denken” en “ontwerp en implementatie”. Hiermee probeert het middelbaar onderwijs deze kloof tussen makers en gebruikers te dichten.

Sommige Europese landen, zoals Estland en Groot-Brittannië, hebben al flinke stappen gezet in het programmeeronderwijs voor kinderen in de basisschool(leeftijd) (Jeuring et al. 2016). Desondanks kampen de leerkrachten over het algemeen met een gebrek aan interesse bij hun leerlingen. Uit mijn eigen ervaringen en die van andere leerkrachten informatica blijkt dat leerlingen vaak aangeven dat ze geen interesse hebben in programmeren. Hier en daar wordt een enkeling wel enthousiast wanneer men een les programmeren aankondigt. Dit blijkt dan meestal wel een leerling te zijn die hiermee al uit eigen beweging bezig geweest is. Men stelt ("Beleef wetenschap!", 2016) dat een gebrek aan interesse bij kinderen en jongeren voor wetenschap niet aan het kind ligt maar aan het de manier waarop de leerstof aan het kind wordt overgebracht.

De eerste programmeertalen waren zeer complex. Over het algemeen waren deze eerder gebaseerd op machinetaal of een reeks van zeer complexe instructies. De nieuwere, hogere programmeertalen zijn dan weer vaak gebaseerd op de Engelse taal. Daardoor is het niet altijd even evident om deze bij te brengen aan niet-Engelstalige kinderen. Er is dus nood aan een meer toegankelijke manier om programmeren tot bij kinderen en jongeren te brengen. Oplossingen hiervoor zijn nu in volle opmars.

Verschillende bedrijven en organisaties ontwikkelen programmeertalen en andere tools, zoals bijvoorbeeld bepaalde types van robots, om spelenderwijs belangrijke programmeerconcepten aan te brengen. Het aanbod blijkt nu echter zo groot te zijn dat leerkrachten het moeilijk hebben om een geschikte tool te kiezen. Deze bachelorproef probeert een antwoord te geven op de vraag welke van deze tools nu efficiënt blijken te zijn.

Hoofdonderzoeksvraag:

Welke invloed hebben educatieve tools op de interesse in het programmeren bij jongeren van 10 tot 14 jaar oud?

Deelvragen:

- Welke educatieve tools zijn geschikt voor de leeftijdsgroep van 10 tot 12 jaar?
- Welke educatieve tools zijn geschikt voor de leeftijdsgroep van 12 tot 14 jaar?
- Wat zijn didactische valkuilen bij het inzetten van deze educatieve tools in de les?

Zowel in de literatuurstudie als in het eigen onderzoek worden heel wat tools geïntroduceerd. Voor sommigen klinken deze benamingen bekend in de oren, anderen zijn

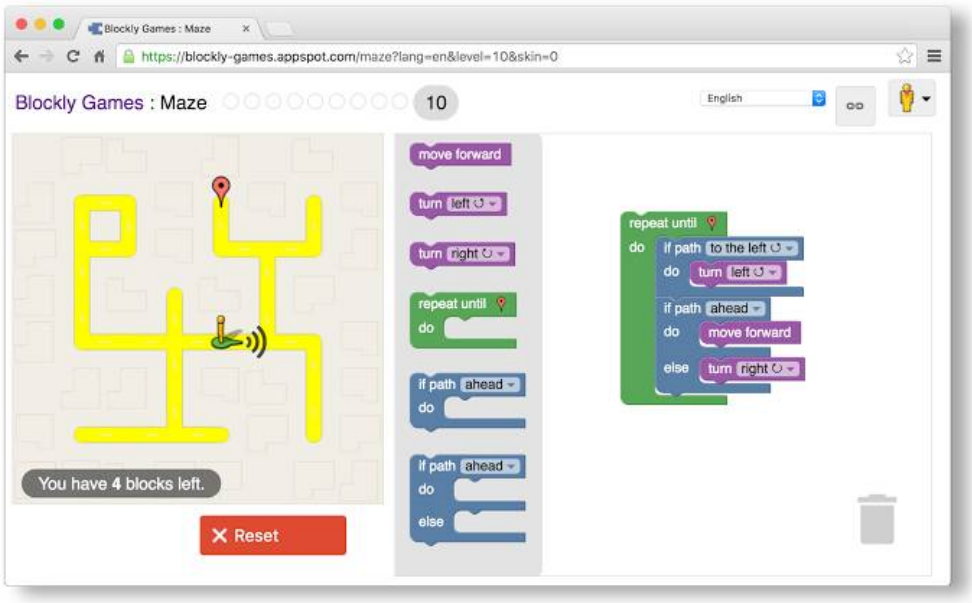
er misschien minder mee bekend. Daarom vindt u hieronder een klein overzicht van tools die doorheen de bachelorproef meermaals vermeld worden.

1 Scratch
Een visuele programmeertaal ontwikkeld op MIT.

2 Lego Mindstorms
Een programmeerbare robot van Lego. Je sluit hier zelf sensoren en motoren op aan door middel van legoblokken.

3 Lego WeDo
Bouw met legoblokken, motoren en sensoren een eigen robot die je door middel van USB op de computer aansluit. Het programma maak je op de computer in een visuele programmeertaal.

4 Blockly
Een visuele programmeertaal ontwikkeld door Google.



5 mBot
Een robotkit die je kan bouwen zoals je zelf wil en die je kan programmeren via een visuele programmeertaal.

6 Arduino
Een simpel elektronica-bord waar je LEDs, sensoren en dergelijke op kan aansluiten.

7 AppInventor
Een webapplicatie om eigen applicaties voor Android te maken aan de hand van een visuele programmeertaal.

I Onderzoek

1 Literatuurstudie

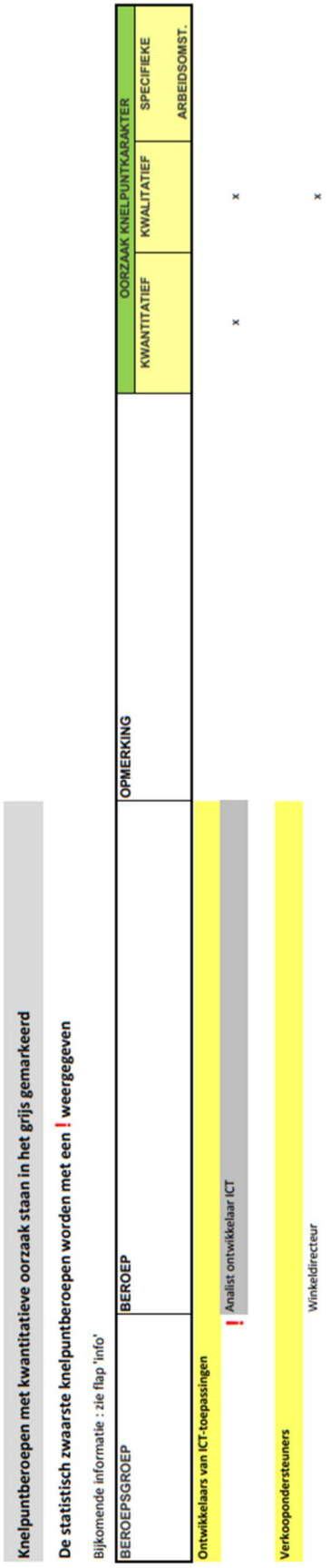
1.1 Waarom dit onderzoek?

Ieder jaar publiceert de Studiedienst van de Vlaamse Dienst voor Arbeidsbemiddeling en Beroepsopleiding een lijst van knelpuntberoepen. Per beroep wordt aangegeven waarom dit een knelpunt vormt. Volgens VDAB (2016) zijn “er drie mogelijke oorzaken:

1. Kwantitatief tekort: te lage uitstroom uit het onderwijs, bijvoorbeeld omdat te weinig studenten die richting kiezen of omdat er geen schoolse opleiding voor bestaat.
2. Kwalitatief tekort: de kandidaten beschikken niet over de gevraagde bekwaamheden.
3. Specifieke arbeidsomstandigheden: weekendwerk, laag loon, zwaar werk, stress, …”

De lijst van knelpuntberoepen (VDAB, 2016, p. 2 - 4) duidt verschillende IT-gerelateerde beroepen aan met zowel een rood uitroepteken als een grijze achtergrond. Een beroep met een rood uitroepteken geeft aan dat het statistisch gezien behoort tot de meest moeilijk in te vullen beroepen. De grijze markering wijst op een kwantitatieve oorzaak. We vinden hier onder meer beroepen als “Bedrijfsanalist ICT”, “Netwerkbeheerder” en “Analist ontwikkelaar ICT” (*Figuur 1*).

Werkgevers in de IT-sector kampen dus met een kwantitatief tekort. Er zijn simpelweg niet genoeg werknemers om al deze jobs in te vullen. VDAB (2016) stelde dat de lage uitstroom uit het onderwijs te wijten kon zijn aan een te lage instroom. Met andere woorden: te weinig studenten kiezen voor een IT-gerelateerde richting. Nochtans lezen we in De Standaard (Van Hollebeke, Van Lent, 2016) dat het aantal inschrijvingen voor IT-gerelateerde richtingen stijgt. In dit artikel werden het aantal inschrijvingen bij hogescholen en universiteiten onder de loep genomen. Bij de hogescholen bleek dat het aantal inschrijvingen voor Toegepaste Informatica met 17% gestegen waren, en voor Elektronica-ICT met 13%. Ook bij UGent is er sprake van een stijging van het aantal inschrijvingen in de opleiding Informatica. Maar liefst 29% meer studenten schreven zich in ten opzichte van het jaar daarvoor. Martine Taeyman van Thomas More (Van Hollebeke, Van Lent, 2016) stelt dat “studenten beseffen bij welke studies de jobkansen het grootst zijn”.



Figuur 1: selectie van twee knelpuntberoepen uit de lijst van VDAB - 2016

Maar hoe komt het dan dat we hier volgens de VDAB met een kwantitatief probleem zitten? De Vlaamse Overheid (2016) stelt ieder jaar een rapport op met het aantal inschrijvingen per opleiding in het hoger onderwijs. Aan de hand van dit rapport kunnen we afleiden hoeveel studenten werkelijk kiezen voor een IT-gerelateerde opleiding.

Er zouden zo'n 234 971 inschrijvingen geweest zijn in het hoger onderwijs (Vlaamse Overheid, 2016, p.13). De richting Toegepaste Informatica zelf had er 4799 (Vlaamse Overheid, 2016, p.17). Dit komt overeen met 2,04% van alle ingeschreven studenten. Ook de richting Elektronica-ICT deed het goed volgens de krant (Van Hollebeke, Van Lent, 2016). Deze opleiding had 1391 inschrijvingen (Vlaamse Overheid, 2016, p. 18), wat neerkomt op 0,6% van het aantal ingeschreven studenten.

De laatste opleiding die het artikel (Van Hollebeke, Van Lent, 2016) aanhaalde was Informatica aan UGent. De inschrijvingen hiervoor stegen met 29%. Volgens het rapport (Vlaamse Overheid, 2016, p. 61-62) heeft Informatica hier 79 inschrijvingen, wat overeenkomt met 0,03%. De richting Wiskundige Informatica in de masteropleiding had 13 inschrijvingen, dat is 0,006%.

Over heel Vlaanderen kozen er 734 studenten (Vlaamse Overheid, p. 21) voor de richting Informatica, wat neerkomt op 0,31% van alle studenten. Wanneer we deze cijfers vergelijken met het aantal inschrijvingen voor bijvoorbeeld Rechten zien we dat er hiervoor 5843 studenten zijn ingeschreven (Vlaamse Overheid, 2016, p. 20). Dat is bijna acht keer zo veel (= 5843/734) als het totale aantal inschrijvingen voor Informatica, terwijl het beroep van advocaat of notaris niet tot de lijst van knelpuntberoepen behoort (VDAB, 2016).

Zoals in de inleiding al aangehaald, is Linda Liukas, de Finse computerprogrammeur en auteur van het boek "Hello, Ruby" bang voor een kloof tussen gebruikers en makers (Nieuws van de Vooruitgang, 2016). Om dit te vermijden zouden we een manier moeten vinden om informatica populairder te maken, zowel in het gebruik ervan als in het maken van deze informatica. “Moet iedereen dan programmeur worden?” vraagt Maas (2015, p. 12) zich af in haar boek “CodeKlas”. Meteen beantwoordt ze haar eigen vraag: “Nee, natuurlijk niet. Maar we omringen ons met steeds meer technologie. Het is handig om wat begrip te hebben van hoe deze technologie werkt. Wanneer je je een voorstelling kunt maken van wat er mogelijk is, wordt het eenvoudiger om met nieuwe technologie om te gaan.”

Is dan de enige reden om lessen programmeren in het onderwijs op te nemen de nood om deze knelpuntberoepen (Vlaamse Overheid, 2016) in te vullen? Hierop geeft Maas (2015, p. 15) het volgende antwoord: “Leren programmeren leert je logisch nadenken, samenwerken, creatief met de computer omgaan, probleemoplossend handelen en projectmatig werken.” Redenen te over dus. Ook in een artikel van EdTechReview (Santosh, 2013) worden nog een rist andere redenen opgesomd waarom het naast de jobmogelijkheden een goede zaak is wanneer we kinderen leren programmeren (uit het Engels vertaald):

- ✓ Het stimuleert creativiteit
- ✓ Het verbetert probleemoplossend denken
- ✓ Het verbetert de analytische vaardigheden
- ✓ Het versterkt logisch denken
- ✓ Het traint vaardigheden in wiskunde

Daarnaast haalt dit artikel (Santosh, 2013) een quote aan van één van de meest bekende personen in IT, Bill Gates: “Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains.“

“Wie niet op een plezierige manier in aanraking is gekomen met moderne technologie, zal minder snel geneigd zijn hier een opleiding in te volgen, er werk in te zoeken of te creëren” (Keulen, Oenen, 2015). We moeten dus volop inzetten om programmeren zo aantrekkelijk

mogelijk te maken. Verschillende bedrijven ontwikkelen materiaal om kinderen spelenderwijs programmeren aan te leren. Of dit materiaal in zijn opzet slaagt, moet getest en onderzocht worden. In deze paper ging ik op zoek naar verschillende studies hierover.

1.2 Wat is programmeren?

Programmeren zien we in deze paper letterlijk als het opstellen van een programma. Dat kan door middel van een programmeertaal zoals Java of C. Het kan ook met een tool of game zoals Scratch of Lego Mindstorms. Het kan tenslotte ook via unplugged programmeren (Jeuring, Corbalan, Montfort, Es, Leeuwestein, 2016, p. 2), wat wil zeggen dat je programmeert zonder gebruik te maken van een computer. Je kan op een speelse manier begrippen uit de informatica aanbrengen.

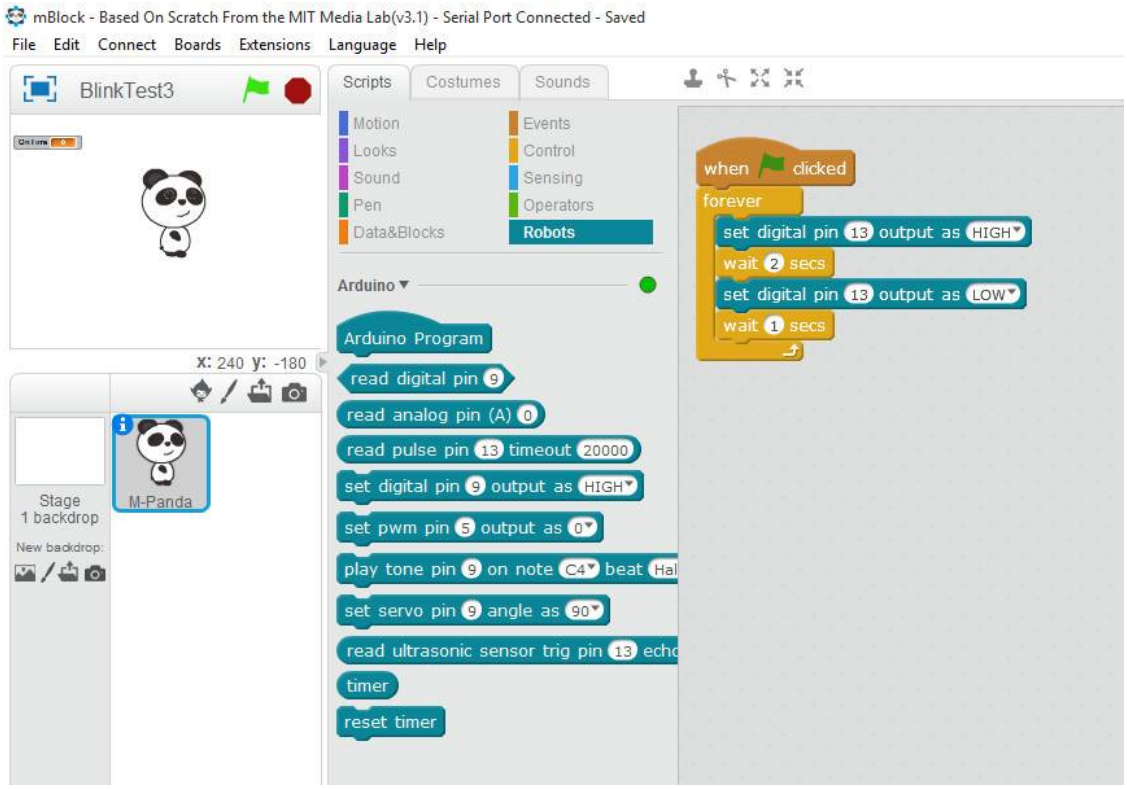
1.2.1 Tekstueel versus visueel

De meest traditionele manier van programmeren is tekstueel programmeren. Om dit te kunnen moet “een leerling de grammatica van de programmeertaal kennen en de namen van de commando’s die specifieke taken uitvoeren” (Jeuring et al., 2016, p. 3). In visueel programmeren daarentegen worden blokjes aan elkaar vastgemaakt. De vorm van deze blokjes zorgt ervoor dat het heel moeilijk tot zelfs onmogelijk is om syntactische fouten te maken. Zo heeft de leerling in een mum van tijd een werkend programma. “De jeugd is buitengewoon visueel ingesteld en gewend aan snelle, heldere informatie. Jongeren interpreteren gemakkelijker beelden dan woorden" (Wijngaards, 2007, p. 4).

1.2.2 Output van visueel programmeren

Het resultaat van visueel programmeren kunnen we gewoonlijk opsplitsen in twee categorieën: beweging op een scherm of fysiek bewegende robots.
Scratch, code.org of mBlock zijn voorbeelden waarmee je beweging op een scherm krijgt. Je werkt daarbij in een virtuele wereld waar je mensen, dieren of andere afbeeldingen kunt manipuleren (figuur 2). Hierbij gebruik je uitsluitend de computer en je eigen verbeelding.

Je kan ook robots zoals Lego Mindstorms of mBot aansturen waarbij je gebruik maakt van visueel programmeren. Vaak zijn deze twee categorieën nauw verweven met elkaar. Zo kan je mBlock gebruiken om een mBot te programmeren.



Figuur 2: programma in mBlock

1.2.3 Motivatie

Een tool kan bijvoorbeeld een programmeerbare robot zijn. Robots spreken tot de verbeelding van leerlingen, ze vinden dit fascinerend. De robot zelf werkt dus motiverend voor deze kinderen (Mayerová, 2012, p. 38; Bertels, Wyffels, Bruneel, 2011, p. 5). Daarnaast bevatten zulke tools ook heel wat mogelijkheden die leerlingen zelf kunnen ontdekken of waarmee ze zelf iets volledig nieuws kunnen ontwerpen. Het is een aantrekkelijke manier om kennis en ervaring op te doen (Mayerová, 2012, p. 38; Kabátová, Pekárová, 2010, p. 2). Leerlingen programmeerconcepten aanleren via tools werkt dus motiverend.

1.3 Wereldwijd

Niet enkel in Vlaanderen zetten enthousiaste leerkrachten zich in om programmeren op een leuke manier aan leerlingen te onderwijzen. Verschillende studies deden onderzoek naar efficiënte manieren om programmeren tot bij kinderen te brengen. In het onderzoek van Jeuring et al. (2016, p. 8) wordt er vermeld dat zij zich baseerden op “artikelen vooral afkomstig uit de VS, maar ook uit Oostenrijk, Bulgarije, China, Duitsland, Griekenland, Slowakije, Zwitserland, Taiwan en Turkije”.

Andere onderzoeken die deze bachelorproef ondersteunen zijn afkomstig uit Nederland (Haverkamp, Van Hout, Leemans, Luksenburg, 2011; Jeuring et al., 2016), België (Bertels, Wyffels, Bruneel, 2011) en Slowakije (Kabátová, Pekárová, 2010; Mayerová, 2012).

1.4 Leeftijden

Jeuring et al. (2016, p. 11-12) onderzocht welke vormen van programmeren bruikbaar zijn bij bepaalde doelgroepen: “Programma’s gevormd door [fysieke] blokken en waarvan de uitvoer wordt getoond met behulp van [bewegende] robots worden enkel gebruikt bij relatief jongere doelgroepen (3-9 jaar)”.

Eerder werd al aangehaald dat deze paper zich focust op visueel programmeren met tools zoals Scratch. Deze vorm is tevens ook de meest gebruikte vorm van programmeren in de bestudeerde onderzoeken (Jeuring et al., 2016, p. 11). “Dit wordt gebruikt door doelgroepen van uiteenlopende leeftijden, waarbij het merendeel van de leerlingen rond de acht jaar en ouder is”.
Deze leeftijdscategorie wordt ook bevestigd door een ander onderzoek (Kabátová, Pekárová, 2010, p. 4) waarbij Lego Mindstorms en Lego WeDo worden ingezet voor het aanleren van programmeren. Deze onderzoekers raden Lego Mindstorms ook aan bij leerlingen vanaf acht jaar en ouder.
Lego WeDo daarentegen vonden ze beter geschikt voor de leerlingen van zeven tot en met elf jaar.

1.5 Begeleiding

Verschillende onderzoeken benadrukken het belang van een goede begeleiding van workshops of lessen om er voor te zorgen dat de leerlingen hier een goede ervaring aan overhouden. Jeuring et al. (2016) vermeldde dat “ondersteuning wordt vaak gepresenteerd als een belangrijke factor voor de motivatie van leerlingen” (p. 12). De hierna volgende vormen van begeleiding kwamen uit de bestudeerde onderzoeken naar voor als stimulans voor de motivatie van de leerlingen. Het is niet voldoende om enkel maar een goede tool te kiezen. Ook de manier waarop je deze tool inzet heeft invloed op de interesse.

1.5.1 Feedback

Wanneer je leerlingen iets aanleert, is het steeds belangrijk hen feedback te geven: wat gaat er al goed, waarop moeten ze nog letten, extra tips, … Feedback helpt het leerproces, maar helpt het ook bij de motivatie? “Ondersteuning in de vorm van feedback (van docenten, experts of medestudenten) draagt bij aan de motivatie” (Jeuring et al., 2016, p. 15).

In mijn onderzoek kijk ik naar de tools die men kan inzetten om programmeren aan te leren. Kunnen deze tools ook een rol spelen in de begeleiding? Jeuring et al. (2016, p. 16) leggen in hun onderzoek uit dat “hoewel programmeren dus ook zonder computers onderwezen kan worden, voegen elektronische apparaten (scherm, robot, geluid of licht) wel aspecten toe die veel leerlingen leuk vinden, en maken ze directe feedback mogelijk.”

1.5.2 Ontdekkend leren

Er zijn verschillende manieren om iemand iets aan te leren. De leraar kan uitleg geven, hij kan de leerlingen opdrachten laten maken, of hij kan de leerlingen op ontdekking laten gaan. Die laatste methode lijkt het meest efficiënt te zijn. Leerlingen beschikken namelijk “over de vaardigheid om onderzoekend en sprongsgewijs door informatie te gaan en toch de betekenis op de nemen. Zij kunnen dit omdat zij ook in staat zijn zoek- en leerdoelen te definiëren” (Wijngaards, 2007, p. 4).

In het onderzoek van Kabátová en Pekárová (2010) wordt er gewerkt met Lego WeDo. De deelnemers krijgen een project dat ze moeten uitvoeren. In hun conclusie merken

Kabátová en Pekárová op dat de deelnemers “niet gemerkt hebben dat ze geen instructies hadden gekregen over hoe ze Lego WeDo-programma’s moesten maken. Meer nog, ze vonden het leuk om dit allemaal zelf te kunnen ontdekken” (p. 7).

Dit blijkt ook een goede manier te zijn om andere programmeertalen aan te leren. Jeuring et al. (2016) raadpleegde voor zijn onderzoek de explorerende studie van Gujberova en Kalas (2013). Dit was een studie met 128 leerlingen, leeftijden van acht tot tien jaar, waaruit bleek dat “ontdekkend leren goed werkt om te leren programmeren” (p. 13).

1.5.3 Team

Wanneer leerlingen in groep moeten werken overleggen ze bij het nemen van bepaalde acties, bespreken ze onderling oplossingen, vragen ze hulp aan elkaar, enzoverder. Werken in een team lijkt ook bij het leren programmeren een gunstig effect te hebben. Het maakt hierbij niet uit of de interactie online gebeurt, zoals bijvoorbeeld via chat, dan wel persoonlijk (Jeuring et al., 2016, p. 16). Overleggen en samenwerken wordt dus sterk aangemoedigd, en de leerlingen waarderen vaak dat ze in team kunnen werken (Kabátová, Pekárová, 2010; Mayerová, 2012). Zelfs Lego raadt op zijn website aan de kits van Lego Mindstorms en Lego WeDo per twee leerlingen te gebruiken.

1.5.4 Project

Kleine opdrachten bij de start van de lessenreeks zijn uitstekend om een tool te leren kennen. Daarna wordt er aangeraden de leerlingen in team een groter project te laten uitwerken (Kabátová, Pekárová, 2010). Uit een onderzoek omtrent “Scratch met 107 leerlingen in groep 7 en 8 [10 tot 12 jaar] blijkt dat project-gebaseerd leren de leerlingen motiveert” (Jeuring et al., 2016, p. 14).

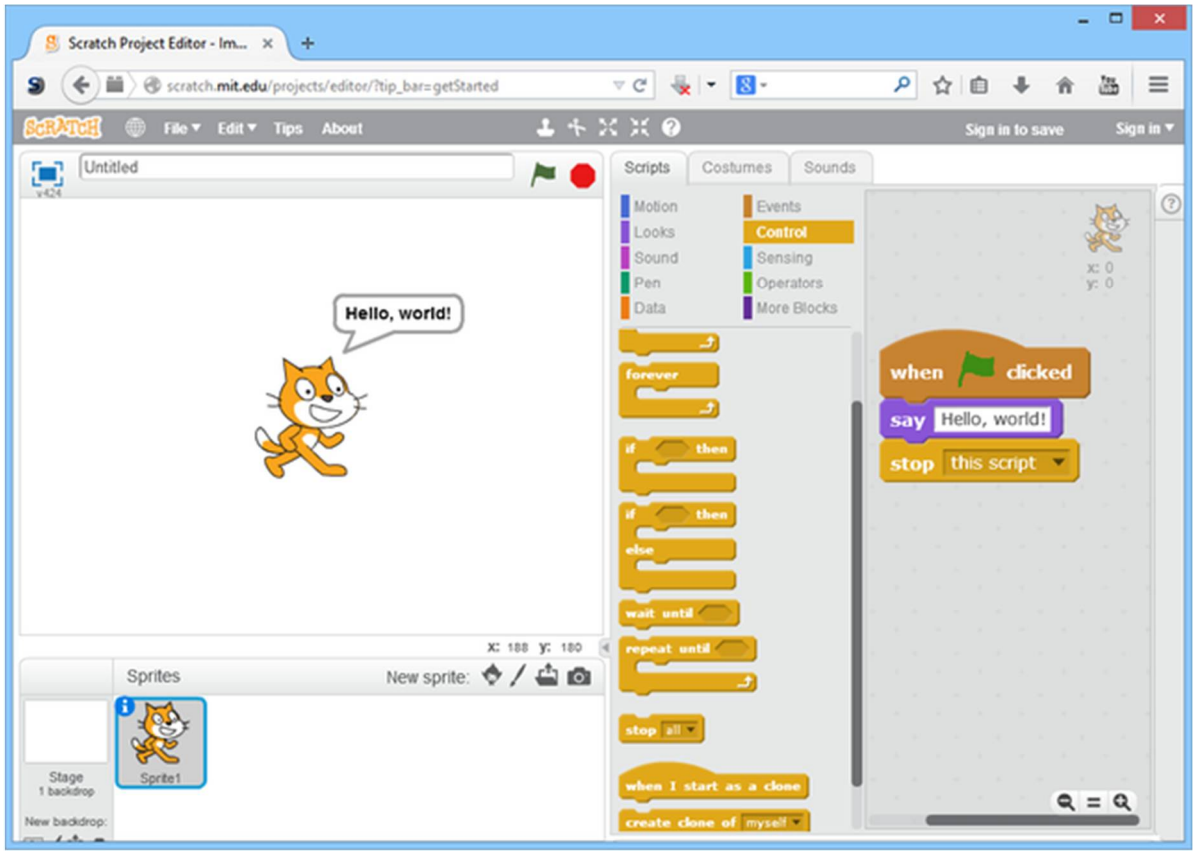
1.6 Gebruikte tools

Aan het einde van deze paper wordt een overzicht gegeven van de tools die vaak met goed resultaat worden ingezet. In verschillende onderzoeken ging ik op zoek naar de gebruikte tools, hoe goed ze het bleken te doen en wat de meest geschikte leeftijdsgroep is voor elk van deze tools.

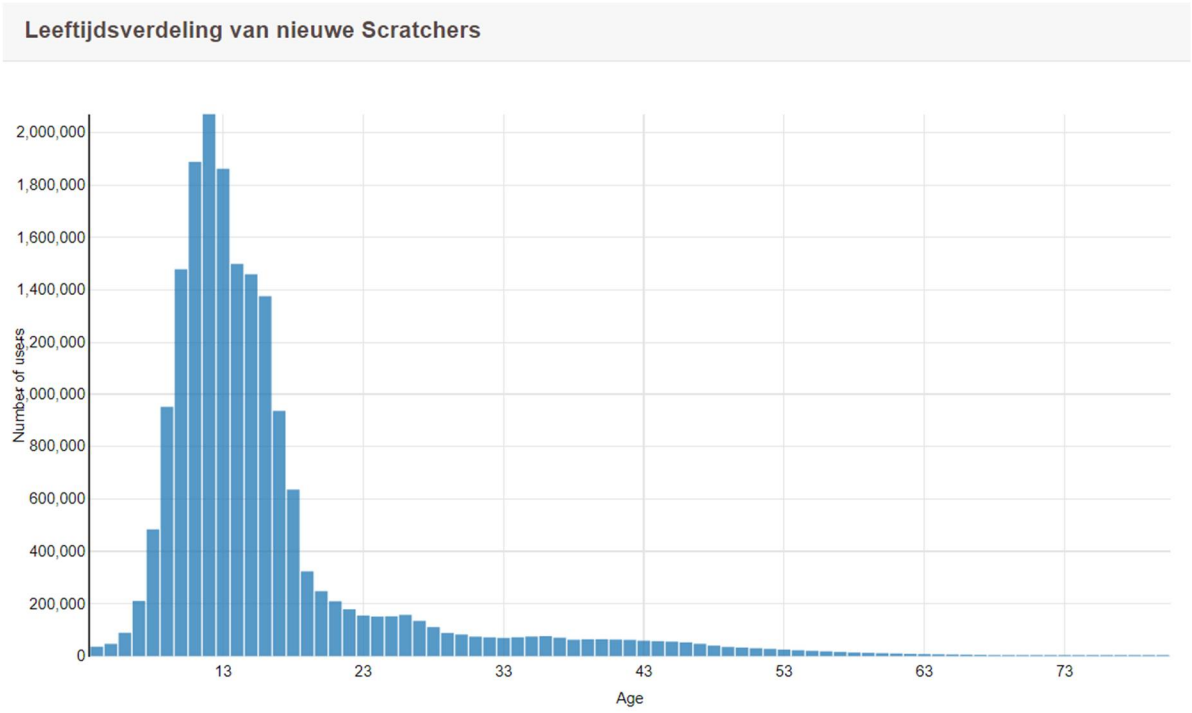
1.6.1 Scratch

Scratch is een grafische programmeertaal waarmee eigen interactieve verhalen, spellen en animaties kunnen gemaakt worden. Deze projecten kunnen tevens gedeeld worden met anderen. Een voorbeeld van een programma is te vinden in *Figuur 3*.

Scratch is speciaal ontworpen voor 8- tot 16-jarigen, maar kan door alle leeftijden gebruikt worden. *Figuur 4* geeft een overzicht van de leeftijden van de mensen die een nieuwe account aanmaakten op de Scratch website. Uit de grafiek volgt dat meer dan twee miljoen gebruikers aangeeft twaalf jaar te zijn.



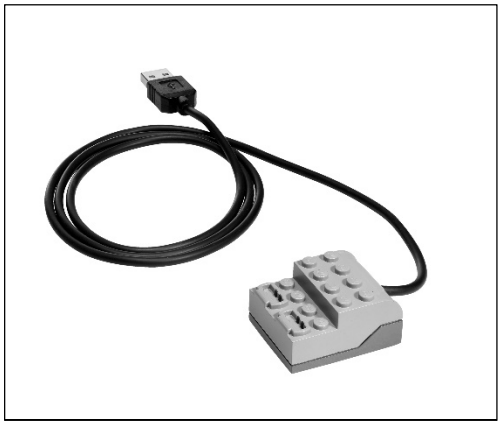
Figuur 3: Voorbeeld van een project gemaakt in Scratch



Figuur 4: Grafiek van scratch.mit.edu, leeftijd van Scratchgebruikers

1.6.2 Lego WeDo

Lego WeDo is een creatie van Lego, de Deense speelgoedfabrikant die gekend is om zijn gekleurde kunstofblokjes en -figuurtjes. Deze blokjes kunnen gecombineerd worden met motoren en sensoren om zo een constructie (*Figuur 6*) te bouwen. De motoren en sensoren worden aangesloten op een zogenaamde Lego USB-hub (*Figuur 5*), die op zijn beurt verbonden wordt met een computer. Met behulp van speciale Lego WeDO software kan deze constructie dan geprogrammeerd worden (*Figuur 7*).
Volgens Lego is deze tool geschikt voor kinderen vanaf zeven jaar.
Volgens Kabátová en Pekárová (2010, p. 4) kan deze tool gebruikt worden tot elf jaar.
Mayerová (2012) gebruikte de tool voor leerlingen van de derde graad van de basisschool (tien en elf jaar).



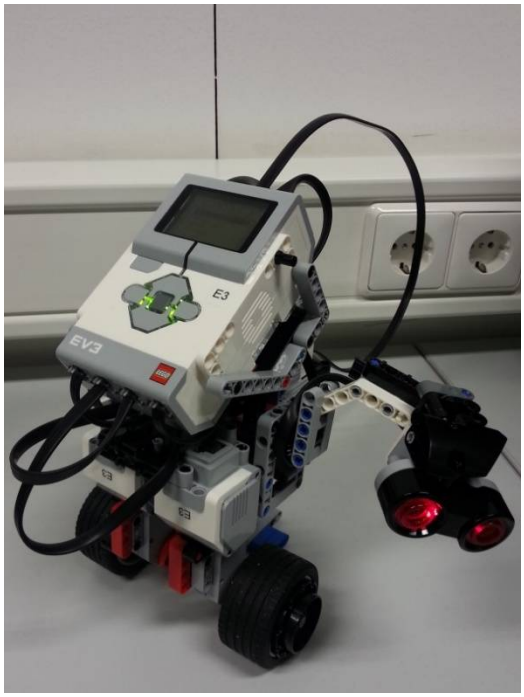
Figuur 5: Lego USB-hub

Figuur 6: voorbeeld van een creatie met Lego WeDo

Figuur 7: voorbeeld van een Lego WeDo programma

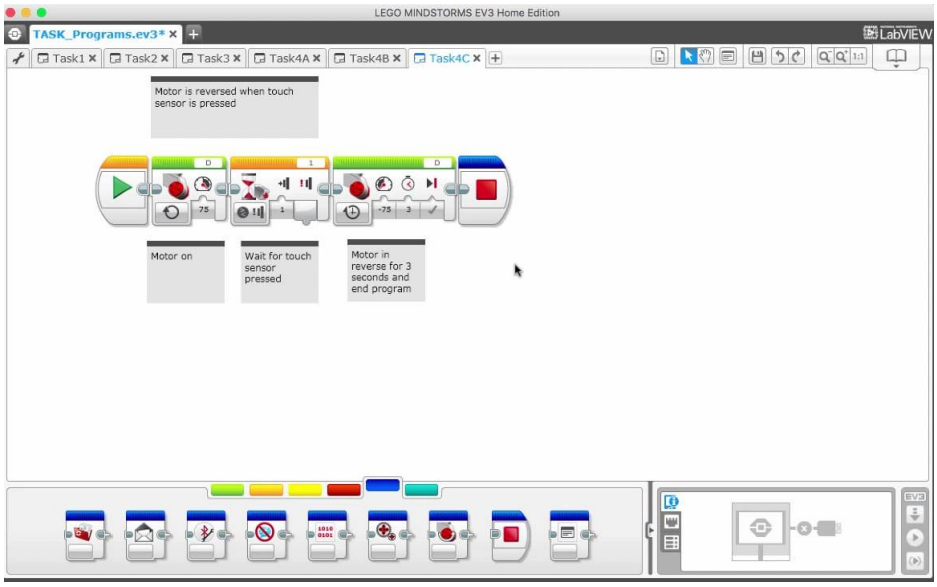
1.6.3 Lego Mindstorms

Een andere tool ontwikkeld door Lego is Lego Mindstorms. Ook met deze tool kunnen eigen creaties gemaakt worden (zie *Figuur 8*) door legoblokken te gebruiken. Centraal in deze tool staat de EV3-blok (*Figuur 9*). Dit is de computer waarop verschillende motoren en sensoren kunnen aangesloten worden om deze vervolgens aan te sturen. In de bijhorende software wordt het programma op een computer ontworpen (*Figuur 10*). Vervolgens wordt dit programma via USB of Bluetooth opgeladen naar de EV3. Doordat deze tool iets ingewikkelder is, raadt Lego aan deze te gebruiken vanaf tien jaar. Kabátová en Pekárová (2010, p. 4) gebruikten in hun onderzoek de tool vanaf acht jaar.



Figuur 8: Robot gemaakt met Lego Mindstorms

Figuur 9: EV3-blok, de programmeerbare computer



Figuur 10: voorbeeld van een programma voor Lego Mindstorms

1.6.4 Dwengo-bord

Dwengo is een non-profit organisatie. Deze organisatie wil zowel jong als oud, zowel leerling als leerkracht motiveren en helpen om te werken met micro-controllers (Bertels, Wyffels, Bruneel, 2011, p. 3). Het door hen ontwikkelde Dwengo-bord bevat een microcontroller met programmeerbaar geheugen, een LCD-display, acht LEDs, vijf drukknoppen, een seriële poort en een USB 2.0 poort voor communicatie met de computer. Dit bord kan ook op een Lego-pakket gemonteerd worden en er kunnen Lego-motoren op aangesloten worden. Het Dwengo-bord wordt geprogrammeerd met Blockly, Ardublock of Labview.

Een recentere versie van dit bord is op de markt gebracht onder de naam Dwenguino, die Arduino-compatibel is.

Figuur 11: Dwengo-bord

2 Onderzoeksopzet

De literatuurstudie toont aan dat grafische programmeertalen en het inzetten van programmeerbare robots de interesse in programmeren bij jongeren verhoogt. Het aanbod aan deze tools is echter zeer groot, daarom bevroeg ik via Google Forms verschillende doelgroepen op hun ervaringen: de deelnemers van de workshops van Progra-MEER, leerkrachten (lager en secundair onderwijs) en coaches van CoderDojo.

2.1 Progra-MEER

Progra-MEER (MEER dan enkel programmeren) is een samenwerkingsverband tussen de universiteiten van Gent, Hasselt, Leuven, medegefinancierd door Google in hun Computer Science for High School programma. Het organiseert verschillende workshops over physical computing en algoritmiek. Het doelpubliek voor deze workshops zijn leerkrachten van de eerste, tweede en derde graad van het secundair onderwijs die lesgeven over informatica, ICT of STEM[1].

In de workshops rond physical computing kregen de deelnemers de kans om met verschillende educatieve tools te leren werken. Het was de bedoeling dat de deelnemers hierna één van deze tools zouden kiezen. Daarmee konden ze dan een project uitwerken dat in de klas kon uitgevoerd worden. Ze konden kiezen uit één van volgende tools:

- ✓ Raspberry Pi
- ✓ Lego Mindstorms met leJos
- ✓ Lego Mindstorms met Labview
- ✓ Dwenguino[2] met Blockly
- ✓ Dwenguino met C++
- ✓ Arduino Uno met ArduBlock

Tijdens de eerste sessie nam ik een enquête af bij de deelnemers om hun beginsituatie in kaart te brengen. Nadat ze alle workshops doorlopen hadden, ging ik na of ze hun project in de klas hadden kunnen uitvoeren en hoe dit verlopen was. Hierbij lette ik vooral op de reactie van de leerlingen en de gebruikte tool.

2.2 Leerkrachten

Ik ging op zoek naar leerkrachten die zelf al enkele educatieve tools kenden en/of deze al in de klas hadden gebruikt. Leerkrachten die ik zelf kende contacteerde ik persoonlijk. Daarnaast postte ik in verschillende facebook-groepen mijn vragenlijst. Zo slaagde ik erin een diverse groep leerkrachten over heel Vlaanderen te bereiken.

Ik vroeg hen in de eerste plaats naar hun eigen ervaringen met tools. Indien ze zelf al één of meerdere tools in hun les hadden gebruikt, vroeg ik hen ook wat de invloed hiervan was op de interesse in programmeren bij hun doelgroep.

2.3 CoderDojo

CoderDojo is een non-profit organisatie die jongeren van zeven tot achtien jaar bij elkaar brengt om in clubverband te leren programmeren. Deze bijeenkomsten worden ondersteund door vrijwilligers (coaches) en zijn bovendien volledig gratis. Onder deze

[1] STEM is een vak waarbij een combinatie wordt gemaakt van science, technology, engineering en math
[2] een open en multifunctioneel Arduino-compatibel microcontrollerbord

vrijwilligers zijn er mensen zowel met als zonder didactische opleiding.
Deze CoderDojo's worden op verschillende plaatsen in België georganiseerd.

In de eerste plaats leren de jongeren programmeren met Scratch. Afhankelijk van de CoderDojo is er een ruimer aanbod aanwezig. Sommige CoderDojo's hebben Arduinosets aangekocht, andere Lego Mindstorms. De keuze van deze tools hangt af van de donaties die die CoderDojo krijgt en de eigen beslissing hoe ze dat geld gaan besteden.

Omdat de Coderdojo's met verschillende educatieve tools werken, contacteerde ik via een forum alle vrijwilligers van CoderDojo Belgium. Daarnaast mailde ik met enkele lead coaches van verschillende CoderDojo's. Ik vroeg hen naar hun ervaringen.

3 Resultaten

Hieronder vat ik de resultaten van de drie hierboven genoemde doelgroepen samen. Alle vragenlijsten en de antwoorden hierop zijn toegevoegd als bijlagen.

3.1 Progra-MEER

Vrijwel alle veertien deelnemers van de Progra-MEER workshops physical computing geven les in wiskunde, fysica, techniek, elektriciteit, informatica of STEM. Slechts één deelnemer geeft momenteel geen les. In de onderstaande tabel vindt u het opleidingsniveau van de verschillende deelnemers.

OPLEIDING DEELNEMERS PROGRA-MEER

Bachelor	4
Master	10
Doctoraat	2
Totaal aantal deelnemers	14

Zes deelnemers hebben drie jaar of minder ervaring in het lesgeven, vijf deelnemers hebben acht tot dertien jaar ervaring en drie deelnemers hebben 30+ jaar ervaring. Van de deelnemers heeft 50% al lesgegeven in programmeren.

Iedere deelnemer had ervaring met programmeren. Uit de reacties op de vraag met welke talen zij al ervaring hadden, bleek een sterk niveauverschil tussen de deelnemers. Sommige hadden enkel een basiskennis van programmeren, maar er zijn ook deelnemers die veel talen kennen. Ongeveer 64% leerde programmeren tijdens de opleiding. 57% verwierf ervaring via zelfstudie. Bijna 36% van de deelnemers nam ook deel aan een workshop over programmeertalen.

Er werd ook gepolst naar de kennis van verschillende tools.
De grote meerderheid, zo'n 71%, heeft geen kennis over Scratch. 21% vindt zichzelf een beginner in Scratch en één deelnemer schat zijn kennis als "middelmatig" in.
Lego Mindstorms is ook bij de grote meerderheid niet gekend, bijna 86%. De overige 14% beschouwt zich als beginner in Lego Mindstorms.
Niemand kende Dwenguino.

Door omstandigheden kon niet iedere deelnemer zijn project in de klas uitvoeren: de leerkracht had op dat ogenblik geen les met de doelgroep van zijn project maar zou het project volgend schooljaar in de klas uittesten; het project was tijdens de workshops niet

volledig afgerond waardoor de leerkracht zich te onzeker voelde om dit in de klas uit te testen; …
Hierdoor heb ik na het aflopen van de workshops slechts vijf van de oorspronkelijke veertien deelnemers kunnen vragen naar hun ervaring.

Deze vijf deelnemers kozen een project waarbij ze gebruik maakten van Arduino Uno en Dwenguino. Zij kozen hiervoor omdat ze dit makkelijk te gebruiken vonden en het meerdere inzetmogelijkheden had. Arduino Uno lijkt dus zowel geschikt voor een beginner als een gevorderde.

Over het algemeen vonden deze vijf deelnemers dat er geen negatieve punten waren aan de Arduino Uno of Dwenguino. Eén deelnemer vond wel dat je beperkt was in stroomvoorziening en een andere deelnemer merkte op dat je veel extra benodigdheden nodig had.

Twee van deze vijf deelnemers hebben hun project in de klas kunnen uitvoeren. Zij gaven aan dat deze les over het algemeen goed verlopen is. Eén deelnemer gaf aan volgende keer dat hij deze les geeft vooraf eerst meer onderzoek te willen doen. Hij vond dat hij er zelf nog niet goed vertrouwd mee was en hierdoor onzeker overkwam. Beide deelnemers meenden wel dat het inzetten van de Arduino Uno de interesse in programmeren bij hun leerlingen had verhoogd.

3.2 Leerkrachten

In totaal vulden 45 leerkrachten mijn vragenlijst in. Twee personen hebben nog geen les gegeven in programmeren.

56,5% leert programmeren aan in de eerste graad en 43,5% in de tweede graad van het middelbaar onderwijs. Ongeveer 28% van de leerkrachten leert dit ook aan in de derde graad van het middelbaar onderwijs.
Een deel van de leerkrachten zet zich in om al in de lagere school programmeren aan te leren. 17,4% doet dit in de eerste vier leerjaren, 28,3% in het vijfde en zesde leerjaar.
Onder de leerkrachten die mijn vragenlijst invulden geeft één persoon les in programmeren in het hoger onderwijs en één persoon aan kleuters.

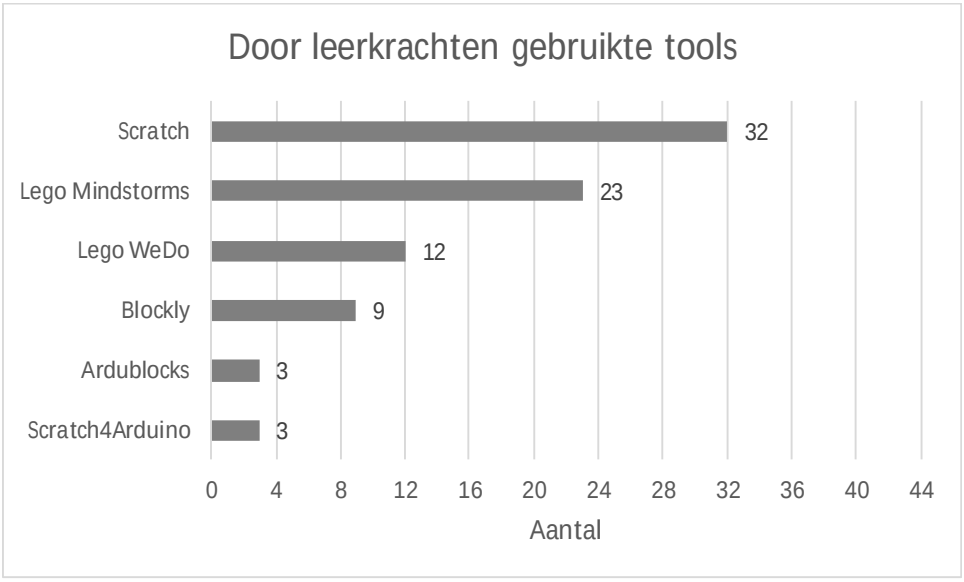
De leerkrachten kregen de vraag in welke onderwijsvormen ze programmeren aanleren. De onderstaande tabel geeft hiervan een overzicht.

Programmeren per onderwijsvorm	
A-STROOM	56,1%
B-STROOM	14,6%
ASO	41,5%
BSO	12,2%
TSO	43,9%
KSO	7,3%

Eén van de 45 leerkrachten kende geen tools zoals Scratch of Lego Mindstorms. De grote meerderheid (89,4%) had zulke tools wel al eens ingezet tijdens de lessen programmeren. Drie bevraagden hadden dit nog niet gedaan, maar twee onder hen willen in de toekomst graag eens een tool in hun lessen gebruiken. Zij waren het erover eens dat dit een leuke afwisseling kan zijn waarin de leerlingen hun creativiteit konden uiten. Eén persoon denkt

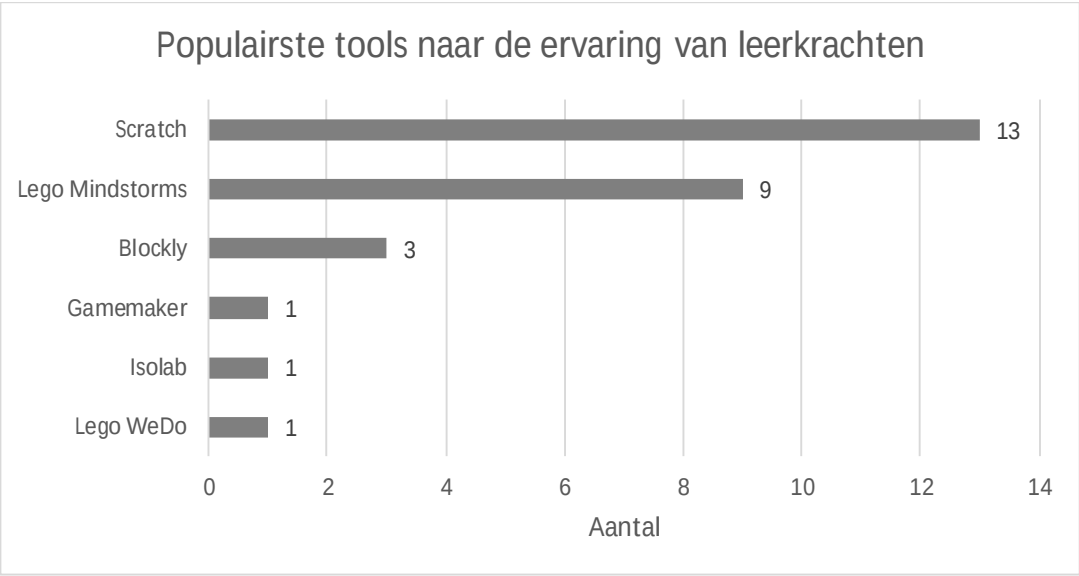
dat de motivatie van de leerlingen hoger zal liggen door deze manier van aanbrengen. De leerkracht die deze tools niet wil gebruiken gaf als reden dat in zijn klassen een hoger niveau van programmeren vereist is dan te bereiken is met deze tools.

De leerkrachten die wel al één of meerdere educatieve tools tijdens de lessen programmeren hadden ingezet (42 van de 45), werden gevraagd op een lijst aan te duiden met welke tools zij al hadden gewerkt. De meest gebruikte tool blijkt Scratch te zijn (72,7%) gevolgd door Lego Mindstorms (52,3%).



Figuur 12: Meest gebruikte educatieve tools in de lessen programmeren

Ik vroeg de leerkrachten ook om hun favoriete tool aan te duiden. 31 personen hebben een antwoord op deze vraag gegeven. Drie personen gaven aan dat hun favoriete tool afhankelijk was van de toepassing of van de doelgroep. Uit de overige 28 antwoorden blijkt dat Scratch de populairste tool is, op de voet gevolgd door Lego Mindstorms.

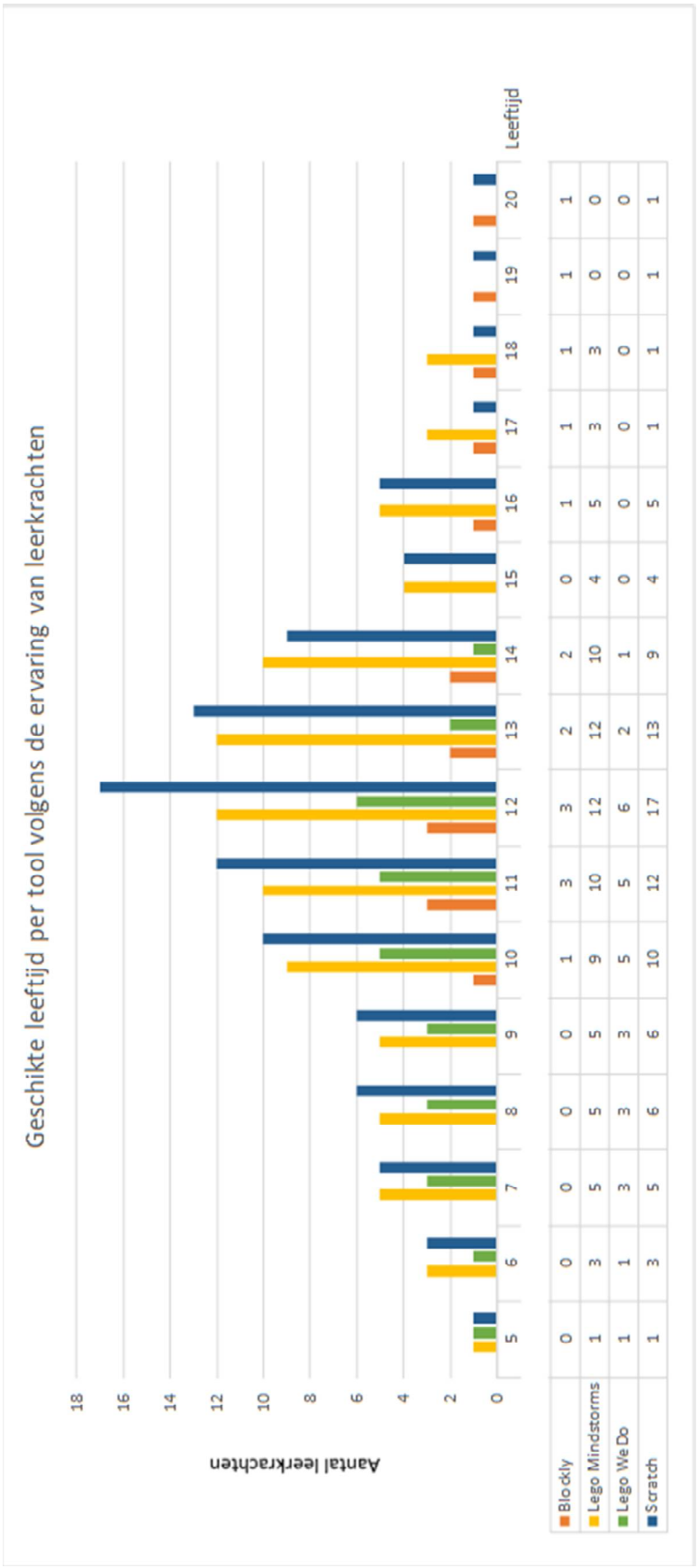


Figuur 13: Populairste tools volgens 28 van de bevraagden

Scratch is dan wel de populairste tool, maar is deze ook geschikt voor elke leeftijd? Ik vroeg de leerkrachten per tool te noteren voor welke leeftijden zij deze, naar hun ervaring, geschikt achtten. De grafiek in *Figuur 14* geeft aan hoeveel leerkrachten een

bepaalde tool geschikt vinden voor een bepaalde leeftijd Om de grafiek niet te overladen nam ik enkel de tools op waarvoor ik meer dan één reactie gekregen had.

Scratch is volgens de leerkrachten bruikbaar voor leerlingen van vijf tot twintig jaar. De tool wordt echter het meest geschikt geacht voor de leeftijdsgroep tien tot veertien jaar.
Ook Lego Mindstorms kan al op vroege leeftijd ingezet worden. Volgens de leerkrachten is dit zeker tot achttien jaar bruikbaar. De ideale leeftijdsgroep voor Lego Mindstorms met de klassieke bijgeleverde software lijkt eveneens tien tot veertien jaar.
Lego WeDo daarentegen is volgens de bevraagde leerkrachten eerder geschikt voor een jongere doelgroep.
Met Blockly bleken weinig leerkrachten ervaring te hebben.



Figuur 14: geschikte leeftijd volgens leerkrachten

In de enquête werd verder aan de leerkrachten gevraagd enkele tips en valkuilen te noteren waarmee je rekening moet houden bij het inzetten van educatieve tools in de lessen programmeren. Volgende tips en valkuilen werden door meerdere leerkrachten aangehaald.

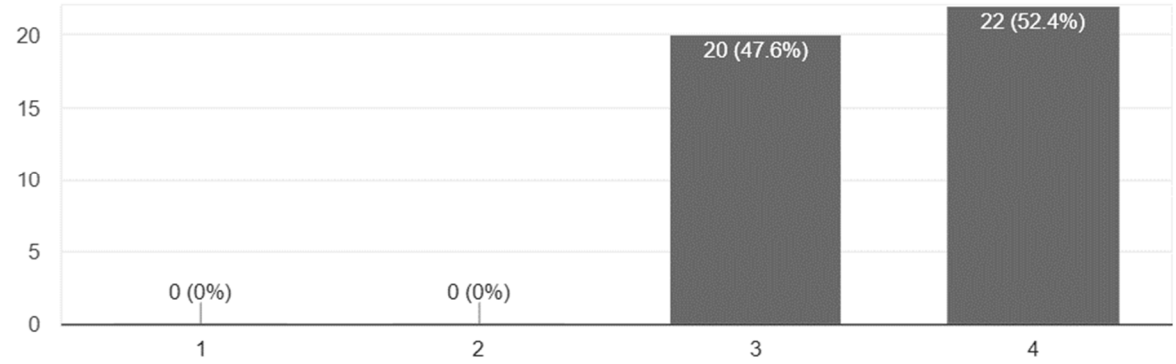
Tips

- ✓ Bereid je goed voor: experimenteer zelf eerst thuis, ontdek de mogelijkheden en de grenzen
- ✓ Bouw stapsgewijs op, structureer, maar laat zelf ontdekken
- ✓ Geef voldoende uitleg en herhaal regelmatig
- ✓ Geef voldoende feedback
- ✓ Zorg voor differentiatie en uitdagingen
- ✓ Laat hen in team werken
- ✓ Voorzie opdrachten voor kinderen die geen eigen ideeën hebben
- ✓ Speel in op de leefwereld van het publiek
- ✓ Laat hen het programma regelmatig opslaan

Valkuilen

- × Te los laten
- × Uitleg/concept te abstract houden
- × Meteen laten programmeren zonder eerst laten nadenken
- × Controle verliezen omdat het té leuk wordt
- × Verwachten dat elke leerling een eigen project kan verzinnen
- × Materiaal kan verloren of stuk gaan
- × Budget: sommige tools zijn zeer duur of hebben extra stukken nodig

In het laatste onderdeel van de enquête moesten de leerkrachten aanduiden in welke mate het inzetten van de tools de interesse in programmeren bij hun leerlingen verhoogd heeft (1: helemaal niet; 4: zeker wel). De vraag luidde als volgt: “Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd?”

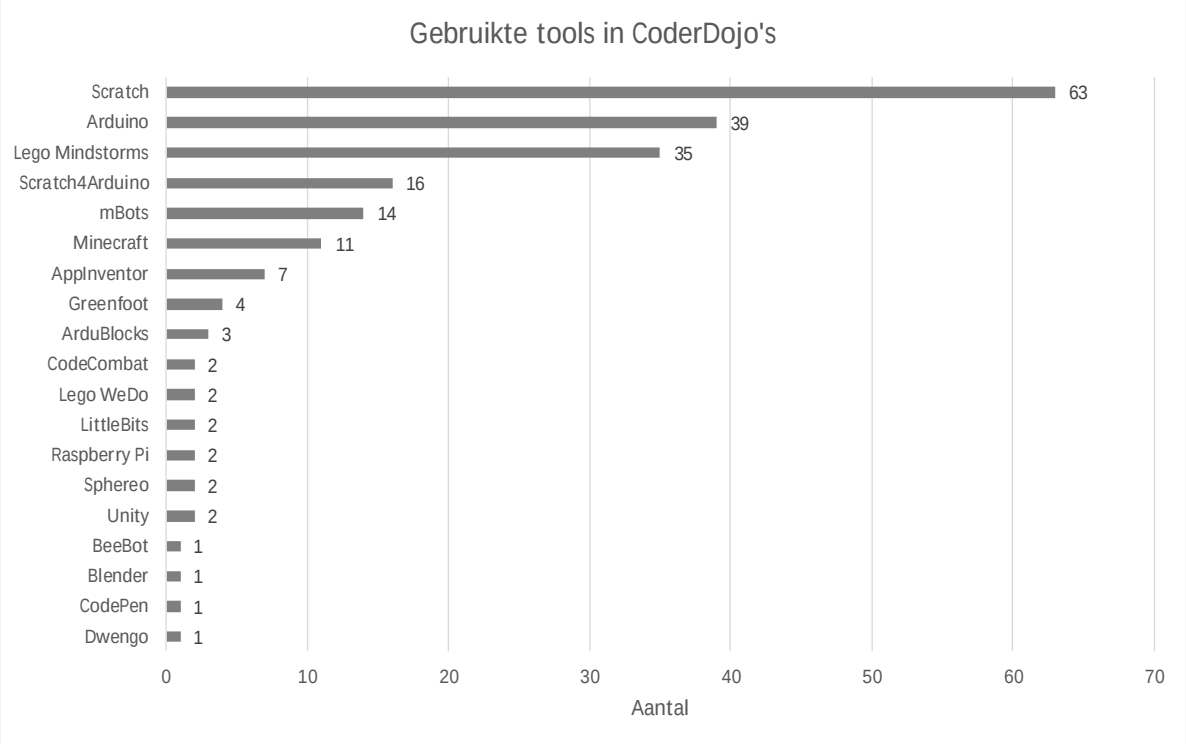


Figuur 15: vraag over de invloed van de tools op de interesse voorgelegd bij 42 leerkrachten

3.3 CoderDojo

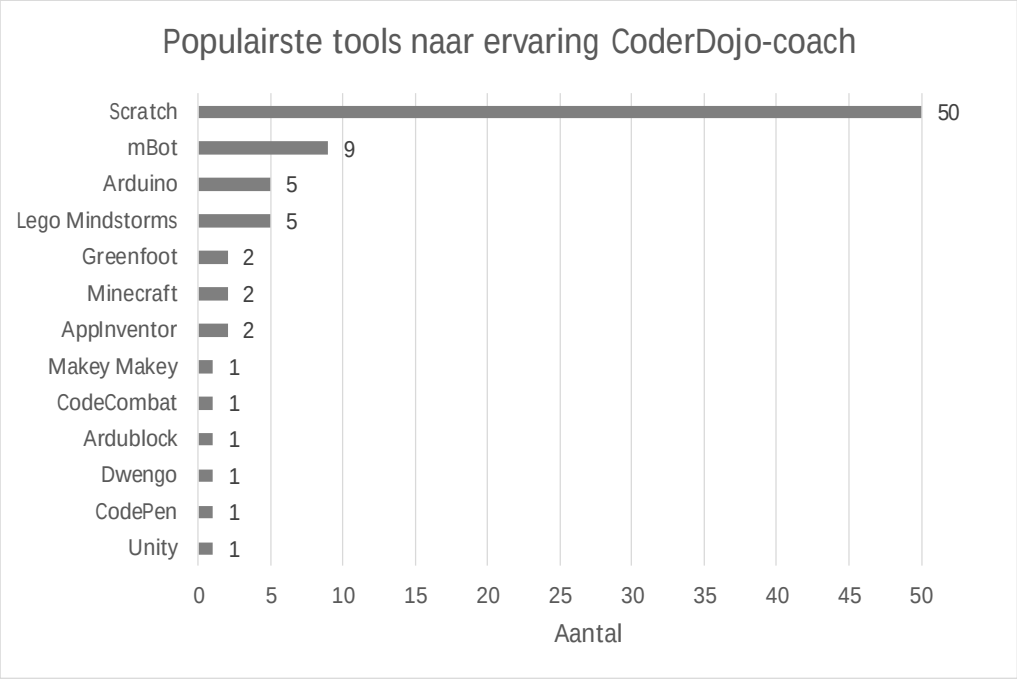
De enquête werd door 63 CoderDojo-coaches ingevuld. 57 coaches gaven aan in welke CoderDojo('s) ze actief zijn. Hieruit blijkt dat de vragenlijst werd ingevuld door coaches uit 40 verschillende CoderDojo's verspreid over Vlaanderen.

De drie meest gebruikte tools in de CoderDojo's zijn Scratch (100%), Arduino (61,9%) en Lego Mindstorms (55,5%). De coaches somden hiernaast nog zestien andere tools op waarmee in hun CoderDojo gewerkt wordt. De verdeling kan je aflezen op *Figuur 16*.



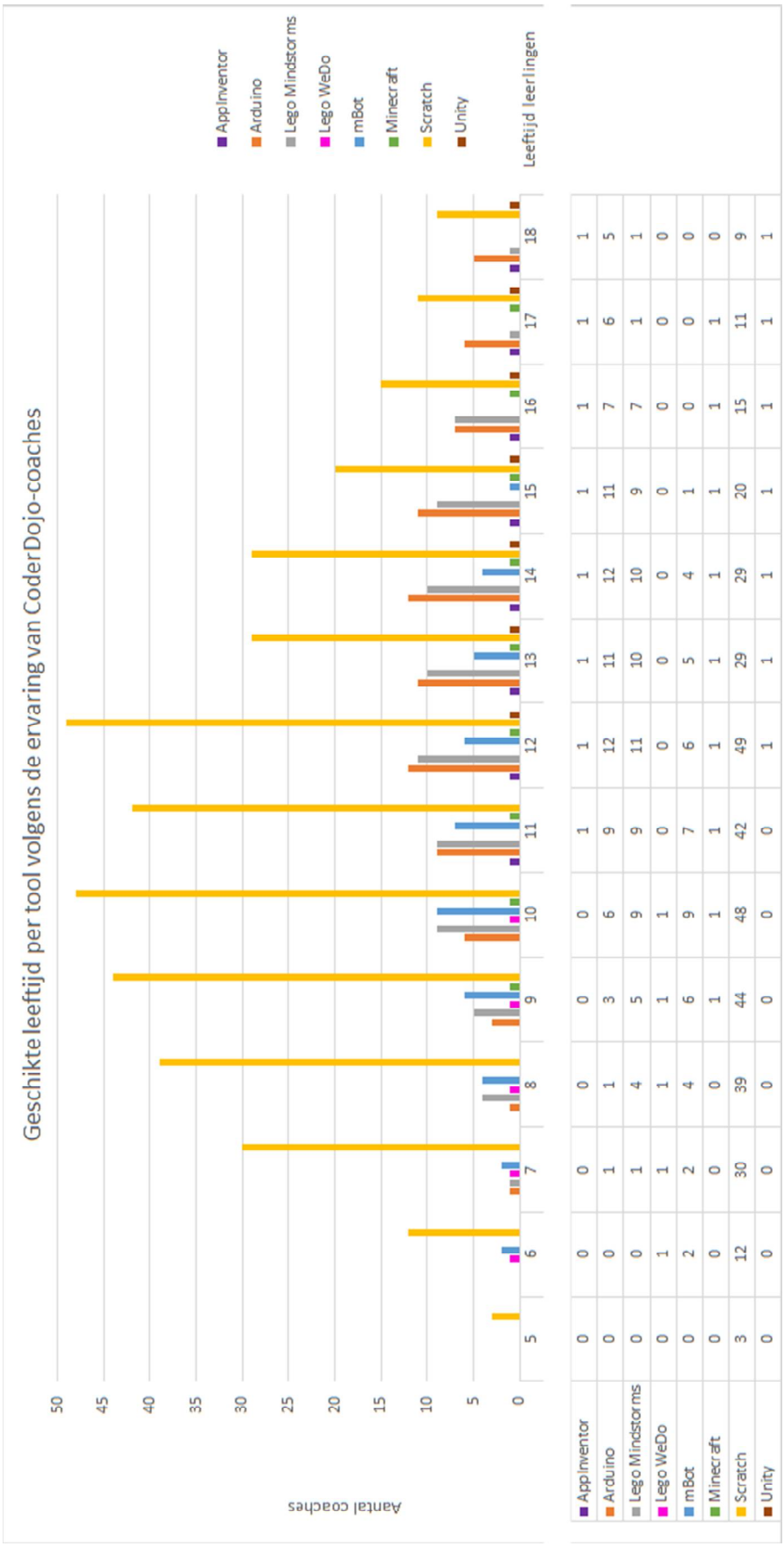
Figuur 16: gebruikte tools door 63 coaches in 40 CoderDojo's

De coaches kregen ook de vraag met welke tools zij de beste ervaringen hebben. De antwoorden hierop zijn terug te vinden in *Figuur 17.*
Scratch blijkt voor 50 van de 63 coaches de meest populaire tool te zijn op basis van positieve ervaringen. Op de tweede plaats kwam de mBot, aangeduid door negen coaches. Vijf coaches duidden Arduino en Lego Mindstorms aan als één van hun favoriete tools. De overige tools werden slechts door één of twee coaches aangeduid.



Figuur 17: populairste tools CoderDojo

Ook hier kregen de deelnemers de vraag per educatieve tool, naar hun ervaring, aan te geven wat zij hiervoor de geschikte leeftijden vonden. *Figuur 18* geeft een overzicht van hun antwoorden.



Figuur 18: geschikte leeftijd CoderDojo

Voor Scratch werd door veel coaches een antwoord gegeven. Twaalf coaches vinden dat men Scratch kan inzetten vanaf zes jaar. Stijgt de leeftijd, dan vinden we meer coaches die achter het gebruik van Scratch op deze leeftijd staan. Voor de twaalfjarigen geven 49 coaches aan dat Scratch geschikt is voor deze leeftijd. Voorbij twaalf jaar daalt het aantal opnieuw.
Voor de andere tools zijn er minder gegevens. Zes coaches vinden tien jaar een geschikte leeftijd voor Arduino, voor elf jaar zijn dat er al negen. Dit loopt zachtjes op tot veertien jaar, en daalt daarna opnieuw.
De mBot is volgens de coaches geschikt van acht jaar tot veertien jaar.
Ook Lego Mindstorms zouden de coaches gebruiken vanaf acht jaar. De coaches vinden dat je deze tool kan inzetten tot zestien jaar.
Bij de overige tools heeft slechts één coach per tool zijn ervaringen omtrent de leeftijd gedeeld. Hierdoor beschouw ik dit niet als representatief.

Om de enquête af te sluiten vroeg ik de coaches enkele tips te noteren voor beginnende coaches. De volgende tips en valkuilen kwamen meermaals terug:

<u>Tips</u>

- ✓ Ga op het internet op zoek naar leuke oefeningen
- ✓ Werk in kleine stapjes
- ✓ Wees zelf goed voorbereid, ken de tool grondig
- ✓ Wees geduldig
- ✓ Moedig hen aan
- ✓ Laat hen zelf experimenteren
- ✓ Voorzie goed ondersteunend leermateriaal (zorgt voor een snelle succeservaring)
- ✓ Geef feedback: tips in plaats van oplossingen
- ✓ Hou rekening met de leefwereld
- ✓ Voorzie uitdagingen

<u>Valkuilen</u>

- × Te abstracte uitleg/opdracht
- × Kinderen maken vaak dezelfde soort programma's als ze:
    - o te los gelaten worden
    - o zelf weinig inspiratie hebben

II Eindbesluit met aanbevelingen en reflectie

4 Eindbesluit

4.1 Hoofdonderzoeksvraag

“Welke invloed hebben educatieve tools op de interesse in het programmeren bij jongeren van 10 tot 14 jaar oud?” was de hoofdonderzoeksvraag voor deze paper. Een antwoord hierop werd gezocht in zowel een literatuurstudie als in mijn eigen gevoerde enquêtes.

De nood is hoog om programmeren interessant te maken voor jongeren. Op de arbeidsmarkt is er een groot tekort aan gekwalificeerde werknemers, simpelweg doordat er te weinig studenten kiezen voor een informatica-gerichte opleiding. Wanneer we programmeren op een aantrekkelijke manier aanbrengen in het onderwijs, kunnen er meer leerlingen geprikkeld worden en geïnteresseerd raken in informatica. Bij het verderzetten van hun studies aan een hogeschool of universiteit zullen er mogelijk meer leerlingen een opleiding in informatica kiezen. Ook diegenen die hier niet voor kiezen, hebben dankzij deze tools toch heel wat programmeerconcepten aangeleerd gekregen. Deze helpen hen bij verschillende vaardigheden zoals probleemoplossend denken en wiskunde. Anderzijds helpen deze hen ook om een beter inzicht te krijgen in de technologieën waarmee we in het dagelijkse leven vaak in aanraking komen, zoals smartphones, tablets of robotstofzuigers.

Educatieve tools bieden een goede oplossing om programmeren aantrekkelijk te maken. Robots spreken tot de verbeelding van jongeren en werken motiverend op zich. De grafische programmeertalen zijn zeer visueel, wat ook stimulerend werkt voor de leerlingen. Ze zijn zo ontworpen dat leerlingen heel snel zien welke blokjes in elkaar passen wat de kans op fouten verkleint. De tools bieden tal van mogelijkheden. Leerlingen kunnen hun creativiteit de vrije loop laten en heel wat nieuwe ontdekkingen doen.

Deze tools zijn niet enkel ontzettend leuk om mee te spelen, ze bieden nog vele andere aspecten om leerlingen te motiveren. Zo is directe feedback mogelijk bij de tools. Leerlingen hoeven zich niet langer blind te staren op waar fouten zitten of te wachten tot de leerkracht tijd heeft om langs te komen omdat de tools meestal duidelijk aangeven waar het probleem zit. Omdat vele tools tevens uitermate geschikt zijn voor teamwerk, is dit ook een stimulans voor de leerlingen. Ze kunnen namelijk met elkaar overleggen en hebben zo niet het gevoel dat ze er alleen voor staan.

De tools laten ook toe hele grote projecten uit te werken zonder dat het te abstract voor de leerlingen wordt. Leerlingen blijken erg gemotiveerd wanneer ze een groter project kunnen uitwerken. Er zijn tal van uitdagingen en mogelijkheden in dergelijke projecten verstopt, zodat de nieuwsgierigheid van de leerlingen geprikkeld blijft. Met andere woorden: de tools blijven hen boeien.

De leerkrachten in het onderzoek merkten op dat het inzetten van zulke tools de interesse in programmeren meestal verhoogde. Geen enkele leerkracht ging hier niet mee akkoord. De leerkrachten gaven daarnaast ook aan dat je deze tools in iedere onderwijsvorm kan inzetten.

4.2 Deelvragen

In deze paper ging ik ook op zoek naar didactische valkuilen wanneer zulke tools in de lessen worden ingezet. Ik ging ook na welke tools geschikt zijn voor de leeftijdsgroepen van tien tot twaalf jaar en van twaalf tot veertien jaar. Alle informatie die ik verzameld heb omtrent deze tools, hun gebruik in de lessen en hun didactische valkuilen heb ik gebundeld in een website: www.edu-tools.weebly.com.

5 Reflectie

Tijdens mijn onderzoek maakte ik verschillende bedenkingen die toch het vermelden waard zijn. Zo zag ik vaak verwijzingen naar Scratch en Lego Mindstorms terugkomen, zowel in de onderzoeken die ik geraadpleegd heb als in mijn eigen onderzoek. Er zijn heel wat tools op de markt, zelfs zo veel dat het als *leek* moeilijk is om te weten welke tool nu eigenlijk het meest geschikt is. Waarom komen dan steeds deze twee tools terug? Een verklaring zou kunnen zijn dat Scratch erg bekend is geworden dankzij de CoderDojo's. Het succes van Lego Mindstorms kan misschien verklaard worden door de grote bekendheid van Lego zelf.

Opmerkelijk vond ik dan ook dat er 35 CoderDojo-coaches aangaven Lego Mindstorms te gebruiken in hun CoderDojo, maar dat er slechts vijf coaches deze kozen als hun favoriete tool. Wil dit dan zeggen dat er betere tools op de markt zijn? Waarom worden deze dan niet gebruikt? Antwoorden op deze vragen zou ik in een verder onderzoek kunnen trachten te vinden. Ik zou hierbij dan eerder gebruik maken van een focusgroep.

Op mijn enquête voor CoderDojo coaches kreeg ik 63 reacties. Uit de vraag in welke CoderDojo('s) ze actief waren bleek dat ik reacties had ontvangen uit 40 verschillende CoderDojo's. Ik vond dit het vermelden waard omdat de resultaten van deze enquête dus veel relevanter zijn dan indien ik slechts uit pakweg vijf verschillende CoderDojo's antwoord had gekregen. Standaard wordt in elke CoderDojo Scratch aangeboden. Voor de rest zijn ze echter volledig onafhankelijk van elkaar en hebben ze de vrijheid te beslissen welke tools ze willen gebruiken. Wel wordt er binnen één CoderDojo ongeveer op dezelfde manier gewerkt. Wanneer de coaches van één CoderDojo dus beslissen dat ze een bepaalde tool inzetten vanaf acht jaar, zullen alle coaches binnen deze CoderDojo dit doen. Indien een bepaalde tool op 40 verschillende plaatsen wordt ingezet vanaf ongeveer acht jaar, is deze informatie natuurlijk veel relevanter dan wanneer deze komt van slechts vijf plaatsen.

Eerder merkte ik al op dat er erg weinig CoderDojo coaches Lego Mindstorms hadden aangeduid als favoriete tool. 50 coaches kozen voor Scratch, terwijl er slechts vijf voor Lego Mindstorms kozen. Bij de leerkrachten is dit verschil veel kleiner. Dertien leerkrachten kozen voor Scratch en negen voor Lego Mindstorms. Binnen mijn onderzoek in het onderwijs blijkt Lego Mindstorms dus populairder te zijn. Een verklaring hiervoor kan zijn dat coaches bij CoderDojo sowieso beginnen met het begeleiden van Scratch. Misschien zijn ze deze tool zodanig gewend dat dit bijna automatisch hun favoriete tool wordt omdat ze deze door en door kennen. Anderzijds kan Lego Mindstorms bij leerkrachten populairder zijn omdat je er minder technische kennis voor nodig hebt. Het is moeilijk om verkeerde schakelingen te maken.

Hoewel ik gefocust was op of educatieve tools de leerlingen motiveerden, ontdekte ik nog heel wat andere stimulansen en voordelen van zulke tools. Groepswerk wordt gestimuleerd door tools. De leerlingen leren dus samenwerken, wat een belangrijke vaardigheid is. Daarnaast leren ze ook werken aan grote projecten. Ik ben ervan overtuigd dat deze punten beginnende leerkrachten goed kunnen helpen. Ik vond het zelf erg interessant de mening van anderen over bepaalde tools te lezen en deze ervaringen onderling te vergelijken.

Graag had ik nog bij de leerlingen zelf gepolst naar hun ervaringen met educatieve tools tijdens de programmeerlessen. Ik wou graag de meningen van die leerlingen horen waarbij de projecten van de deelnemers van Progra-MEER werden uitgeprobeerd. Jammer genoeg waren heel wat van deze projecten nog niet uitgevoerd, waardoor ik te weinig data zou hebben om tot representatieve resultaten te komen. Dit zou in een volgend onderzoek wel aan bod kunnen komen.

Een laatste belangrijke reflectie bij mijn eigen gevoerde enquêtes is dat de gevormde conclusies gebaseerd zijn op de antwoorden van een kleine doelgroep. Deze aantallen zijn niet groot genoeg om de antwoorden representatief te maken voor heel Vlaanderen.

Bibliografie

Bertels, P., Wyffels, F., Bruneel, K. (2011). *Bouw een intelligente robot in de klas met Dwengo.* Geraadpleegd op 13 januari 2017, van https://biblio.ugent.be/publication/1954732

Haverkamp, D., Van hout, E., Leenmans, S., & Luksenburg, B. (2011). *Beeld van informaticaonderwijs op de middelbare school*. Geraadpleegd op 8 oktober 2016 van http://repository.tue.nl/716278

Het Nieuws van de Vooruitgang. (2016, 18 maart). Meer interesse in programmeren. Geraadpleegd op 20 december 2016, van http://hetnieuwsvandevooruitgang.nl/2016/03/meer-interesse-programmeren/

Jeuring, J., Corbalan, G., Van Montfort, J., Van Es, N., & Leeuwstein, H. (2016). *Vorm en effect van het programmeeronderwijs in het primair onderwijs*. Geraadpleegd op 8 oktober 2016, van https://www.semanticscholar.org/paper/Vorm-En-E-ect-Van-Programmeeronderwijs-in-Het-Jeuring-Corbalan/29756ce085c090d19183cddcf7ffa0f9b83048e5

Kabátová, M., Pekárová, J. (2010). *Learning how to teach robotics.* Geraadpleegd op 8 oktober 2016, van http://lcd.ent.sk/docs/kabatova_pekarova_lego_constructionism2010.pdf

Keulen, H., van., Oenen, S., van. (2015). *'Wat je kunt verzinnen, kun je ook maken!', 3D-printen op basisscholen in Almere.* Gedownload op 13 januari 2017, van http://www.iederkindeentalent.nl/wp-content/uploads/2015/11/Rapport-3D-printen-op-basisscholen-in-Almere-2015-1.pdf

Maas, P. (2015). *CodeKlas*. Groningen, Nederland: Boektweepuntnul.

Mayerová, K. (2012). *Pilot Activities: LEGO WeDo at Primary School.* Geraadpleegd op 8 oktober 2016, van http://www.terecop.eu/TRTWR2012/trtwr2012_submission_05.pdf

Onderwijs van de Vlaamse Gemeenschap. (2012) *Leerplan secundair onderwijs Toegepaste informatica.* Geraadpleegd op 11 augustus 2017, van http://pro.g-o.be/blog/documents/2012-048.pdf

Santosh, B. (2013). Why should young students be encouraged to learn programming? *EdTechReview.* Geraadpleegd op 22 juli 2017, van http://edtechreview.in/trends-insights/insights/486-young-students-encouraged-to-learn-programming

Snoeck, M. (2002). *Waarom kiezen meisjes niet voor informatica?* [Paper]. Geraadpleegd op 21 november 2016, van https://lirias.kuleuven.be/bitstream/123456789/122351/1/monique_snoeck.pdf

Vanhollebeke, A., Van Lent, R. (2016, 12 september). Welke studierichtingen zijn het populairst? *De Standaard.* Geraadpleegd op 22 juli 2017, van http://www.standaard.be/cnt/dmf20160912_02464399

Vlaamse Dienst voor Arbeidsbemiddeling en Beroepsopleiding. (2016). *De knelpuntberoepenlijst van 2016 – 2017.* Geraadpleegd op 22 juli 2017, van https://www.vdab.be/trends/vacatureanalyse.shtml

Vlaamse Dienst voor Arbeidsbemiddeling en Beroepsopleiding. (2016). *Lijst van knelpuntberoepen in Vlaanderen 2016.* Gedownload op 22 juli 2017, van https://www.vdab.be/trendsdoc/vacatureanalyse/knelpuntberoepenlijst2016.pdf

Vlaamse Overheid. (2016). *Hoger onderwijs in cijfers. Aantal inschrijvingen op 31 oktober 2015. Academiejaar 2015 – 2016.* Geraadpleegd op 22 juli 2017, van https://issuu.com/vlaanderen-be/docs/48b9da58-cf90-417c-90be-ddc9d0d16d3e

Vlaams Verbond van het Katholiek Secundair Onderwijs. (2011). *Leerplan secundair onderwijs Informatica tweede graad ASO-KSO-TSO.* Geraadpleegd op 11 augustus 2017, van http://ond.vvkso-ict.com/leerplannen/doc/Informatica-2011-039.pdf

Wtnschp / Wetenschap in en uit Brussel. (z.d.). *Beleef wetenschap!* Geraadpleegd op 21 december 2016, van http://www.wtnschp.be/aanbod/jongeren/

Wijngaards, G. (2007). *Innovatief leren met jongeren.* Gedownload op 13 januari 2017, van https://www.inholland.nl/media/10075/essay-or-guus-wijngaards-2.pdf

Bijlagen

6 Enquêtes via Google Forms

Alle antwoorden op de vragenlijsten vindt u op de volgende bladzijdes. Onderstaande tabel geeft aan op welke pagina's u de antwoorden kan vinden per doelgroep.

De *timestamp* identificeert de gebruiker.

Progra-MEER	Beginsituatie: p. 43 tot 45 Evaluatie project: p. 46 tot 47
Leerkrachten	p. 48 tot 65
CoderDojo	p. 66 tot 75

6.1 Progra-MEER

Timestamp	Welke onderwijsvakken geeft u?	Welke opleiding heeft u gevolgd? (master/licentiaat - bachelor)	In welke school geeft u les?
12/9/2016 9:27:40	Fysica stem	Master	Guldensporencollege kortrijk
12/9/2016 9:27:45	Wiskunde informatica	Master	Tihf brugge
12/9/2016 9:28:57	Frans, Informatica, STEM	licentiaat Romaanse talen	Guldensporencollege Harelbeke
12/9/2016 9:30:12	Techniek	Master	Hogeschool Gent
12/9/2016 9:32:19	Progammeren	Licenciaat	Sint-Bavo
12/9/2016 9:32:22	Elektriciteit-elektronica	Hoger onderwijs lange type=master	Sint-Laurens Zelzate
12/9/2016 9:33:19	Elektriciteit	Bachelor	Don Bosco Haacht TSO/BSO
12/9/2016 9:35:12	Toegepaste Informatica	Bachelor secundair onderwijs	VLOT! Lokeren
12/9/2016 9:35:39	Toegepaste informatica, toepassingspakket, multimedia, netwerken, project, databanken,	Bachelor	Go! Atheneum en leefschool de tandem
12/9/2016 9:36:01	Wiskunde, STEM	Master in de wiskunde (+doctoraat in de wiskunde)	Stella Matutina College Lede
12/9/2016 9:38:17	TV elektriciteit, TV schakeltechnieken, TV Engineering elektriciteit	Burgerlijk elektrotechnisch ingenieur	Don Bosco Sint-Denijs-Westrem
12/9/2016 9:39:05	TW Elektriciteit, STEM, Elektronica, ICT	Master	Don Bosco Haacht
12/9/2016 12:56:47	Geen	A3	Atheneum merelbeke
12/9/2016 12:58:35	Wiskunde	Master	Popelin Lyceum Merelbeke

Timestamp	In welke onderwijsvormen heeft u al les gegeven?	Hoeveel jaar geeft u al les?	Heeft u al ervaring met programmeren?	Indien u al ervaring heeft: In welke programmeertalen heeft u ervaring?
12/9/2016 9:27:40	ASO	37	Ja	Ooit Basic Pascal fortran vba Java php c for arduino mindstorm niet altijd
12/9/2016 9:27:45	TSO, KSO	13	Ja	Ver verleden c++
12/9/2016 9:28:57	ASO	31	Ja	beetje JAVA Greenfoot, beetje python
12/9/2016 9:30:12	Hogeschool	13	Ja	Fortran
12/9/2016 9:32:19	ASO	0.5	Ja	Visu al Basic, SQL,
12/9/2016 9:32:22	TSO	10	Ja	Flowcode
12/9/2016 9:33:19	BSO, TSO	30	Ja	Arduino (programmeren in C)
12/9/2016 9:35:12	ASO, TSO	3	Ja	vb.net, C#
12/9/2016 9:35:39	ASO, BSO, TSO	8	Ja	Java
12/9/2016 9:36:01	ASO, Universitair	8	Ja	C++, Java
12/9/2016 9:38:17	BSO, TSO	3	Ja	Java, C, C++, Flowcode, VHDL, SystemC, assembler (behalve Flowcode is het allemaal wel al lang geleden, 10-tal jaar)
12/9/2016 9:39:05	TSO	2	Ja	
12/9/2016 12:56:47	Volwassen onderwijs vdab	1	Ja	.net Java php
12/9/2016 12:58:35	ASO	1	Ja	Python, Java, C, C++

Timestamp	Indien u al ervaring heeft: Waar heeft u deze ervaring verworven?	Heeft u al eerder lesgegeven in programmeren?	Heeft u al ervaring met Dwenguino?	Heeft u al ervaring met Scratch?	Heeft u al ervaring met Lego Mindstorms?
12/9/2016 9:27:40	In mijn opleiding, In een workshop/module/nascholing rond programmeren, Zelfstudie	Ja	1	1	2
12/9/2016 9:27:45	In mijn opleiding	Nee	1	1	1
12/9/2016 9:28:57	In een workshop/module/nascholing rond programmeren, Zelfstudie	Ja	1	2	1
12/9/2016 9:30:12	In mijn opleiding	Nee	1	1	1
12/9/2016 9:32:19	Zelfstudie	Nee	1	3	1
12/9/2016 9:32:22	In een workshop/module/nascholing rond programmeren	Nee	1	1	1
12/9/2016 9:33:19	In een workshop/module/nascholing rond programmeren, Zelfstudie	Ja	1	1	1
12/9/2016 9:35:12	In mijn opleiding	Nee	1	2	1
12/9/2016 9:35:39	In een workshop/module/nascholing rond programmeren, Zelfstudie	Nee	1	2	2
12/9/2016 9:36:01	In mijn opleiding, Tijdens doctoraat	Nee	1	1	1
12/9/2016 9:38:17	In mijn opleiding, Zelfstudie	Ja	1	1	1
12/9/2016 9:39:05	In mijn opleiding, Zelfstudie, werknemer	Ja	1	1	1
12/9/2016 12:56:47	In mijn opleiding, Zelfstudie	Ja	1	1	1
12/9/2016 12:58:35	In mijn opleiding	Ja	1	1	1

Timestamp	Welke van de tools die u zag in de workshops heeft u ingezet of zou u graag inzetten tijdens uw les? Gelieve er slechts een te noteren.	Waarom kiest u voor deze tool? Wat zijn volgens u de voordelen?	Ondervond u ook negatieve punten van deze tool? Welke?	Heeft u uw project in uw klas kunnen uitvoeren?
6/14/2017 19:29:15	Arduino/dwenguino	Van beginner tot gevorderde, iedereen kan hiermee aan de slag. Ook veel info beschikbaar via het web.	Neen	Neen
6/16/2017 21:38:33	Arduino	Uitdaging, nog niet eerder mee gewerkt / vrij eenvoudig	Beperkt in stroomvoorziening	Neen
6/29/2017 10:41:16	Arduino	Gemakkelijk bruikbaar. Leerlingen kunnen van thuis gemakkelijk programmeren, zonder extra kosten.	Neen	Neen
6/29/2017 21:57:35	Arduino Uno	Keuze v/d vakwerkgroep	Neen	Ja
7/4/2017 20:59:24	Dwenguino/arduino	Handig te gebruiken, leuk concept, meerdere inzetmogelijkheden	Veel extra benodigdheden	Ja

Timestamp	Hoe is uw les verlopen? (op een schaal van 1-4)	Welke problemen ondervond u? Dit kunnen didactische problemen zijn of problemen met de tool zelf.	Wat zou u anders doen moest u deze les nog eens geven?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
6/14/2017 19:29:15				
6/16/2017 21:38:33				
6/29/2017 10:41:16				
6/29/2017 21:57:35	3	nvt	nvt	3
7/4/2017 20:59:24	4	Zelf nog niet altijd even goed mee vertrouwd	Vooraf meer onderzoek	4

6.2 Leerkrachten

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
3/24/2017 13:58:29	Ja	Eerste graad middelbaar, Tweede graad middelbaar	A-stroom	Ja	Ja	
3/24/2017 14:09:51	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	
3/24/2017 23:46:01	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	
3/25/2017 14:56:59	Ja	Derde graad middelbaar	TSO	Ja	Nee	Nee
3/25/2017 15:09:31	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), Eerste graad middelbaar, Tweede graad middelbaar, Derde graad middelbaar	A-stroom, ASO, TSO	Ja	Ja	
3/25/2017 15:10:05	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), Eerste graad middelbaar, Tweede graad middelbaar	A-stroom, ASO, TSO	Ja	Ja	
3/28/2017 12:51:14	Ja	Eerste graad middelbaar	ASO	Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
3/24/2017 13:58:29		Scratch		12-13 jarigen (1ste middelbaar)	Lego Mindstorms
3/24/2017 14:09:51		Scratch			
3/24/2017 23:46:01		Scratch, Lego Mindstorms	Scratch	12-14 jaar	Lego Mindstorm
3/25/2017 14:56:59	Omdat ik les geef in de derde graad in zeer specifieke informaticarichtingen (Informaticabeheer en boekhouden-informatica). Voor deze richtingen is een hoger niveau van programmeren vereist voor later in het werkveld.				
3/25/2017 15:09:31		Scratch, Lego WeDo, Alice, studio code, osmo, Lego fix the factory, lightbot, verborgen, probot	Geen specifieke voorkeur, het hangt af van de doelgroep	6j - 16j	Reeds opgesomd
3/25/2017 15:10:05		Scratch, Lego Mindstorms, Lego WeDo	Lego WeDO	10 - 12 jaar	Scratch, Lego Mindstorms
3/28/2017 12:51:14		Scratch		eerste graad	/

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
3/24/2017 13:58:29	Eerste graad	Spendeer voldoende aandacht aan het leren programmeren en niet enkel aan het maken van de toepassing(en)	De leerlingen vergeten vaak dat ze aan het programmeren zijn en zien het soms meer als spelletjes spelen	3
3/24/2017 14:09:51				
3/24/2017 23:46:01	14-16 jaar	Neem tijd om alles uit te leggen, laat ze zelf exploreren maar niet te los laten, geef structuur en voldoende feedback, maak het tof en luchtig	Snel afgeleid (iets anders programmeren), spelletjes spelen in de tool zelf, afwijken van de vragen, te veel hooi op de vork willen nemen	4
3/25/2017 14:56:59				
3/25/2017 15:09:31	Sommige al vanaf 6j	Leefwereld van de jongeren. Linken leggen met het echte programmeren. Concrete voorbeelden geven	Te abstracte uitleg	3
3/25/2017 15:10:05	10 tot 14 jaar	Het goed begeleiden van de activiteiten! Je kan de leerlingen niet volledig los laten hier in. Voorzie ook uitdagingen voor kinderen die geen ideetjes hebben.	Niet elk kind is even creatief en sommige hebben geen eigen uitdagende ideetjes.	4
3/28/2017 12:51:14		voldoende herhalen, kleine testjes zodat ze de vaardigheden onder de knie krijgen	weinig tot geen voorkennis van de leerlingen. Je moet ze ook voldoende warm maken voor het concept omdat programmeren vaak als abstract wordt gezien	4

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
3/28/2017 19:30:34	Ja	Tweede graad middelbaar	ASO	Ja	Ja	
3/28/2017 21:39:39	Ja	Eerste graad middelbaar, Tweede graad middelbaar	A-stroom, B-stroom, ASO, TSO	Ja	Ja	
4/3/2017 19:09:19	Ja	Eerste graad middelbaar, Tweede graad middelbaar	International School	Ja	Ja	
4/3/2017 19:31:16	Ja	Lagere school (5de en 6de leerjaar), Eerste graad middelbaar	A-stroom	Ja	Ja	
4/3/2017 20:11:09	Ja	Lagere school (5de en 6de leerjaar), Eerste graad middelbaar	A-stroom	Ja	Ja	
4/3/2017 20:12:53	Ja	Eerste graad middelbaar	A-stroom, ASO	Ja	Ja	
4/3/2017 20:23:10	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	
4/3/2017 22:17:02	Ja	Eerste graad middelbaar	A-stroom, TSO	Ja	Ja	
4/3/2017 22:22:38	Ja	Derde graad middelbaar	ASO	Ja	Ja	
4/3/2017 22:34:42	Ja	Tweede graad middelbaar, Derde graad middelbaar	ASO	Ja	Ja	
4/3/2017 23:20:34	Ja	Lagere school (5de en 6de leerjaar)	Primair onderwijs	Ja	Ja	
4/4/2017 0:39:26	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
3/28/2017 19:30:34		Scratch		15jaar	Ken geen andere
3/28/2017 21:39:39		Scratch	scratch	13-14	
4/3/2017 19:09:19		Scratch, Lego Mindstorms	Scratch	14-16 jaar	Lego Mindstorm, Sphero
4/3/2017 19:31:16		ozoblockly		10-14	
4/3/2017 20:11:09		Lego Mindstorms, Code.org	Lego mindstorms	5e 6e middelbaar + 1ste graad	Code.org
4/3/2017 20:12:53		Lego Mindstorms, Blockly, Ardublocks	Blockly	12-14	Lego mindstorms
4/3/2017 20:23:10		Scratch, Lego Mindstorms, Blockly	Blockly	1ste graad SO	
4/3/2017 22:17:02		Scratch, Scratch4Arduino, Lego Mindstorms, Lego WeDo, Blockly	Dat is afhankelijk van de toepassing.	12-14 jaar	mBlock
4/3/2017 22:22:38		Lego Mindstorms		17-18	Python
4/3/2017 22:34:42		Lego Mindstorms		2e en 3e graad	
4/3/2017 23:20:34		Scratch, Codekinderen, beebot, mbot	Scratc	Groep 8 10 tot 12 jaar	Beebot
4/4/2017 0:39:26		Lego Mindstorms, Blockly, Ozoblockly	De variatie maakt het interessant...	12-14	

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
3/28/2017 19:30:34	Vanaf 13jaar wordeen onze lln hiermee vertrouwd gemaakt	eerst logisch redeneren en dan programmeren	Te snel willen programmeren ipv te redeneren	3
3/28/2017 21:39:39		Structuur en goede uitleg van het programma	te moeilijke oefeningen	4
4/3/2017 19:09:19	11-16 jaar	Geef structuur mee, zorg voor differentiatie, maak de oefeningen tof (inspelen op hobbies, trends, ...)	Afwijken van de voorziene leerstof, niet te luchtig overgaan, voorbereid zijn op moeilijke vragen	4
4/3/2017 19:31:16	10-14	Geef de kinderen ook een Nederlandstalige uitleg/instructiefiche	Sommige kinderen gebruiken enkel de blokken die ze echt kennen en proberen niets nieuws te verkennen.	3
4/3/2017 20:11:09	Alle leeftijden	Korte duidelijke instructie en lln laten experimenteren	Duur materiaal dat gemakkelijk verloren gaat	4
4/3/2017 20:12:53	12	Basis goed uitleggen. Per 2 lln 1 robot	Batterij, stukken, veel voorkennis vanuit lagere school, verschillende beginsituatie	3
4/3/2017 20:23:10		Zelf ook kennis hebben		3
4/3/2017 22:17:02	12 tot 16	Zorg dat je het zelf onder de knie hebt.	Verouderde computers op school	4
4/3/2017 22:22:38	Alle	Sterk differentiëren, tempo van de leerlingen verschilt heel sterk		3
4/3/2017 22:34:42		Analyse - niet te veel zelf willen verklappen - foutenanalyse laten uitvoeren		3
4/3/2017 23:20:34	Alle	Waar je handig materiaal kunt vinden. Niet het wiel uitvinden!	Zelf willen kunnen voordat de kinderen het doen.	4
4/4/2017 0:39:26		Het zelf volledig beheersen van de tools	Ozobot met ozoblockly doet niet steeds zijn ding... fouten ... gevoelige sensors...	3

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
4/4/2017 8:49:19	Ja	Eerste graad middelbaar, Tweede graad middelbaar	Plusklassen	Ja	Ja	
4/4/2017 8:53:11	Ja	Derde graad middelbaar, Hoger onderwijs (universiteit/hogeschool)	TSO, Bachelor Toegepaste Informatica	Ja	Nee	Ja
4/4/2017 9:10:44	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	
4/4/2017 9:56:59	Nee	Eerste graad middelbaar	A-stroom, B-stroom	Ja	Ja	
4/5/2017 19:19:45	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar)		Ja	Ja	
4/5/2017 20:16:37	Ja	Eerste graad middelbaar	A-stroom, ASO, TSO	Ja	Ja	
4/9/2017 10:26:02	Ja	Tweede graad middelbaar, Derde graad middelbaar	ASO, KSO	Ja	Ja	
4/10/2017 19:12:19	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), kleuters		Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
4/4/2017 8:49:19		Lego Mindstorms		10-16 jaar	Raspberri Pi
4/4/2017 8:53:11	Het is een leuke afwisseling, en kan helpen bij leerlingen die de logica nog niet volledig snappen. Het zou alles wat concreter kunnen maken				
4/4/2017 9:10:44		Scratch, Lego Mindstorms	Lego mindstorms	13/14 jaar	scratch
4/4/2017 9:56:59		Scratch, Lego Mindstorms, Lego WeDo, Ardublocks	Lego Mindstorms	10-14	Lego WeDo
4/5/2017 19:19:45		Scratch, Lego Mindstorms, Lego WeDo	Lego Mindstorms	lagere school	Scratch
4/5/2017 20:16:37		Scratch, Lego Mindstorms, Edison	Mindstorm	12_14	Edison
4/9/2017 10:26:02		Isolab	Isolab	ongeveer 15 jaar	Weet ik niet
4/10/2017 19:12:19		Scratch, Lego Mindstorms, Lego WeDo	Momenteel heb ik meest ervaring met scratch die ik geef in de lagere school en scratch junoir in het 3de kleutersklas en 1ste graad basisschool	3de kleuterklas en lagere school	indien ik tijd heb verzend ik dit per e-mai

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
4/4/2017 8:49:19	10-18		Complexiteit van systemen, omslachtigheid van lego mindstorms (eerste versies) Programma's die ofwel té simpel zijn en weinig variatie of vrijheid aanbieden, ofwel zijn dz programma's té complex en is de leercurve zo stijl dat studenten snel afhaken.	4
4/4/2017 8:53:11				
4/4/2017 9:10:44	13/14/15 jaar	basisuitleg en dan zelf laten ontdekken met cursus	kinderen denken snel dat ze het kunnen maar lopen vast op opgelegde opdracht	4
4/4/2017 9:56:59	10-12	Duidelijke stappenplannen voor in het begin alsook de leerlingen elk onderdeel van de tool duidelijk laten maken. Bijvoorbeeld: De luidspreker bij Lego Mindstorms ook even demonstreren dat dat logo voor geluid staat.	Dat de leerkracht zelf niet 100% alles onder de knie heeft.	4
4/5/2017 19:19:45	lagere school	/	/	3
4/5/2017 20:16:37	9-14			3
4/9/2017 10:26:02	14-15 jaar	Dat ze de structuren goed kennen en begrijpen.	Het komt niet altijd zinnig over voor de leerlingen.	3
4/10/2017 19:12:19	/	Bij de jongste eerst kennismaken met symbolen en het concreet uitvoeren (met het lichaam), nadien verkennen van scratch (je kan dit met een verhaal doen) zodat de jongere kinderen gewoon meedoen		4

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
4/11/2017 1:04:03	Ja	Eerste graad middelbaar	A-stroom	Ja	Ja	
4/18/2017 18:52:28	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), Eerste graad middelbaar	A-stroom, B-stroom, ASO, BSO, TSO	Ja	Ja	
5/2/2017 11:22:19	Ja	Tweede graad middelbaar	TSO	Ja	Ja	
5/2/2017 12:12:15	Ja	Eerste graad middelbaar, Tweede graad middelbaar	A-stroom, ASO	Ja	Ja	
5/2/2017 12:54:20	Ja	Tweede graad middelbaar, Derde graad middelbaar	TSO	Ja	Ja	
5/2/2017 13:01:03	Ja	Tweede graad middelbaar, Derde graad middelbaar	ASO, BSO	Ja	Ja	
5/2/2017 13:46:25	Ja	Tweede graad middelbaar, Derde graad middelbaar	TSO	Ja	Ja	
5/2/2017 14:48:42	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar)	basisschool	Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
4/11/2017 1:04:03		Lego Mindstorms		1e middelbaar	
4/18/2017 18:52:28		Scratch, Scratch4Arduino, Lego Mindstorms, Lego WeDo, Blockly	Scratch en Lego Mindstorms	3e graad lager	mBlock
5/2/2017 11:22:19		Scratch, Lego Mindstorms	Scratch	4e jaar secundair (16j)	
5/2/2017 12:12:15		Scratch, Visual Studio	Moeilijk te zeggen vanwege de beperkte lestijden die hieraan besteed konden worden. mBlock (waar een robot 'mBot' geprogrammeerd wordt met scratch - mBlock is dus in feite bijna scratch) werkt zéér motiverend. Alleen stel ik mij de vraag in hoeverre ze bewust bepaalde leerinhouden verwerken.		Visual Studio => reden: vanaf een 2e graad ASO lijkt mij dit nog altijd een misschien wel minder 'leuk' maar wel zéér relevant programma. Het simuleert immers sterk hoe het programmeren er later aan toe gaat. We mogen niet de fout maken programmeren zodanig leuk te willen maken dat er dan niet voldoende uitgediept worden. Beter 5 oprecht gemotiveerde en sterke programmeurs dan 10 sterk gemotiveerde maar minder sterke programmeurs.
5/2/2017 12:54:20		Scratch, BlueGriffon	BlueGriffon	16j	
5/2/2017 13:01:03		Code.org		16-20 jaar	Code.org
5/2/2017 13:46:25		Scratch, Alice	Alice	15	Alice
5/2/2017 14:48:42		Scratch, Lego Mindstorms, Lego WeDo, Blockly	Lego Mindstorms	6e leerjaar	Ozobot

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
4/11/2017 1:04:03		Werk met uitdagingen, laat kinderen zelfs ontdekken. Zeker geen voorgekauwde kost!	Gevoelig hardware, bv sensoren	4
4/18/2017 18:52:28	10-14 jaar	Voldoende achtergrondkennis + durven experimenteren en op zoek gaan mét leerlingen	De verscheidenheid wanneer je ze door mekaar gebruikt.	4
5/2/2017 11:22:19	3e graad lagere school	Simpel beginnen, programma's maken die aansluiten bij hun leefwereld en interesses	Te moeilijk beginnen, logisch denken is voor sommige leerlingen makkelijker dan voor anderen.	4
5/2/2017 12:12:15	Scratch, mBlock e.d.: 5+6e lj basisonderwijs en 1ste graad secundair.	Dat alle iteraties duidelijk besproken worden en ook andere belangrijke principes. Het mag niet teveel een spel lijken of indien dat wel zo is moet wel op de één of andere manier nagekeken worden in hoeverre ze bepaalde principes en kennis verwerven.	Sommige zijn vrij eenvoudig, waardoor het soms te speels lijkt.	3
5/2/2017 12:54:20		afhankelijk van de leeftijd van de kinderen		3
5/2/2017 13:01:03	12-20			3
5/2/2017 13:46:25	14	Vaak wil je onmiddellijk starten in het programma en het probleem aanpakken. Het is echter erg belangrijk eerst het probleem (vraagstuk) goed te analyseren en eerst op papier een korte analyse zetten. De vraag (probleem) moet opgedeeld worden in kleinere delen.	De leerlingen willen te snel werken en onmiddellijk in het programma 'prutsen''	3
5/2/2017 14:48:42	van kleuter tot adolucent	start eenvoudig, zeg niet te veel		4

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
5/2/2017 15:20:02	Ja	Derde graad middelbaar	TSO	Ja	Ja	
5/2/2017 20:34:22	Ja	Gamedesign kamp	A-stroom, B-stroom, ASO, BSO, KSO, TSO	Ja	Ja	
5/3/2017 7:43:42	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), Eerste graad middelbaar, Tweede graad middelbaar, Derde graad middelbaar	A-stroom, TSO	Ja	Ja	
5/3/2017 14:12:53	Ja	Eerste graad middelbaar, Tweede graad middelbaar	A-stroom, B-stroom, ASO, BSO, KSO, TSO	Ja	Ja	
5/4/2017 22:11:25	Ja	Lagere school (5de en 6de leerjaar)		Ja	Ja	
5/6/2017 17:42:50	Ja	Tweede graad middelbaar	TSO	Ja	Ja	
5/7/2017 11:10:06	Ja	Tweede graad middelbaar	ASO	Ja	Ja	
5/8/2017 16:53:05	Ja	Lagere school (1ste tot 4de leerjaar), Lagere school (5de en 6de leerjaar), Eerste graad middelbaar, Tweede graad middelbaar, Derde graad middelbaar	A-stroom, B-stroom, ASO, BSO, TSO	Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
5/2/2017 15:20:02		Scratch, Blockly, Code studio	Blockly	16-20	Scratch
5/2/2017 20:34:22		Scratch, Blockly, Gamemaker	Scratch naarmate oudere kinderen gamemaker	Scratch 5e 6e leerjaar gamemaker 1e tot 6e middelbaar	Gamemaker
5/3/2017 7:43:42		Scratch, Lego Mindstorms, Lego WeDo	Scratch	7-14	Blockly
5/3/2017 14:12:53		Scratch, Lego Mindstorms, Codecademy	Lego mindstorms	1e graad a en b	Arduino
5/4/2017 22:11:25		Scratch, Blockly	Scratch	11 jarigen	Ozoblockly
5/6/2017 17:42:50		Scratch		14-15 jaar	Lego Mindstorms
5/7/2017 11:10:06		Scratch			
5/8/2017 16:53:05		Scratch, Lego Mindstorms, Lego WeDo, Scratch jr	Scratch	vooral Lager Onderwijs (L2-L6) een 1ste grad Secundair	Absoluut

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
5/2/2017 15:20:02	12-20	Algoritmisch denken stimuleren en aanleren. (Al is het maar door de medeleerlingen commando's te laten geven aan een leerling die een parcours moet afleggen. Is voor mij telkens een actieve instap van het onderwerp programmeren.)	Blockly leert leerlingen zeker algoritmisch denken, scratch is voor mij al een stap verder.	4
5/2/2017 20:34:22	14+	Het visuele aspect + regelmatig laten testen + tussenversie laten opslaan	Bij 5e 6e leerjaar weten sommige leerlingen niet hoe ze met een computer moeten werken	4
5/3/2017 7:43:42	7-13	Leren redeneren en computationrel denken ipv knopjes klikken	Verval naar spelletjes spelen ipv leren programmeren	3
5/3/2017 14:12:53	2e en 3e graad	Werkende voorbeelden laten zien. Tonen wat mogelijk is.	Lego mindstorms doet niet exact wat je programmeert. Voor leerlingen met ass soms lastig	4
5/4/2017 22:11:25	Minimum 10 jaar	Zelf door en door kennen	Zelf niet genoeg beheersd	4
5/6/2017 17:42:50	13-16 jaar	Interesse lln, budget	Veelheid van verschillende oplossingen om tot de oplossing van het probleem te komen. Leerlingen die focussen op een oplossing hebben het hiermee moeilijk.	3
5/7/2017 11:10:06				4
5/8/2017 16:53:05	L1: Scratch JR, vanaf L2 : WeDo, vanaf L5 : Mindstorms	Wees niet bang dat je niet alles weet, laat de kinderen zelf experimenteren	- Teveel uitleg willen geven - Bij robots is er veel enthousiasme en dan is het soms moeilijk de groep terug onder controle te krijgen wat nodig is om meer dan 1 ding te kunnen aanleren	3

Timestamp	Geeft u les, of heeft u les gegeven in programmeren?	In welk jaar leert u programmeren aan?	In welke onderwijsvormen onderricht u programmeren?	Kent u grafische tools zoals Lego Mindstorms, Ardublocks of Scratch?	Heeft u al grafische tools, zoals Lego Mindstorms of Scratch ingezet tijdens de lessen programmeren?	Zou u graag een van deze tools inzetten in uw lessen?
5/19/2017 11:36:37	Nee			Neen	Nee	Ja
5/19/2017 19:48:25	Ja	Lagere school (5de en 6de leerjaar), Eerste graad middelbaar	A-stroom, TSO	Ja	Ja	

Timestamp	Waarom zou u eerder wel of niet een tool willen inzetten tijdens uw lessen?	Welke tools heeft u al (regelmatig) ingezet om kinderen te leren programmeren?	Indien u meer dan één tool al inzette tijdens uw les, met welke tool had u dan de beste ervaring?	In welke leeftijdsgroep had u deze tool(s) gebruikt?	Welke andere tools vindt u ook zeker een aanrader?
5/19/2017 11:36:37	Die tools stellen je in staat de leerstof stapgewijs en op een leuke manier aan te brengen. De motivatie van de leerlingen zal hoger liggen omwille van de speelsere manier van aanbrengen. Ze zien ook meer concreet wat ze doen.				
5/19/2017 19:48:25		Scratch, Scratch4Arduino, Lego WeDo, Ardublocks	Scratch	1e graad	Pas

Timestamp	Voor welke leeftijdsgroep zou u deze tools aanraden?	Als men kinderen wil laten programmeren met één van bovenstaande tools, waarop moet men dan, volgens u, voornamelijk letten (tips voor beginnende leerkrachten)?	Wat heeft u als valkuilen ervaren bij het gebruik van zo'n tools?	Heeft het inzetten van zulke tools de interesse in programmeren bij uw leerlingen verhoogd? (op een schaal van 1-4)
5/19/2017 11:36:37		Maak het niet té plezant.		4
5/19/2017 19:48:25	Pas	De taal	Niet te diep geneste functies	3

6.3 CoderDojo

Timestamp	In welke CoderDojo(s) bent u actief?	Welke tools worden er in uw CoderDojo gebruikt?	Met welke tool(s) heeft u de beste ervaringen?
5/1/2017 20:37:54	Vilvoorde	Scratch, Arduino	Scratch
5/1/2017 20:38:44	Aalst en Brussel1082	Scratch, Lego Mindstorms	Scratch
5/1/2017 20:40:30		Scratch, Lego Mindstorms, Mindcraft, AppInventor	Scratch en Lego Mindstorms
5/1/2017 20:41:11	Antwerpen Centrum, Antwerpen Permeke en Deurne	Scratch, Scratch4Arduino, Arduino	Scratch, mBots
5/1/2017 20:42:28	Aalter	Scratch, Lego Mindstorms, Arduino	Scratch
5/1/2017 20:44:12	Turnhout	Scratch, Lego Mindstorms, mBlock	Scratch
5/1/2017 20:49:06	Dendermonde	Scratch, Unity	Unity
5/1/2017 21:01:33	Sint-Niklaas	Scratch, Arduino, Minecraft	Scratch
5/1/2017 21:06:32		Scratch	Scratch
5/1/2017 21:06:44	CoderDojo Waterloo	Scratch	Scratch
5/1/2017 21:27:35	Lubbeek	Scratch, CodePen	Scratch, CodePEN
5/1/2017 21:40:05	Gent	Scratch, Lego Mindstorms, Arduino, Java	ze lopen alle 4 vol
5/1/2017 21:42:14	Genk	Scratch, Arduino, Minecraft	Scratch
5/1/2017 21:44:47	klein brabant	Scratch, Lego Mindstorms, Lego WeDo	scratch mbot
5/1/2017 21:46:09	UHasselt	Scratch, Lego Mindstorms, AppInventor	AppInventor, Scratch
5/1/2017 22:01:15	Geel	Scratch, mBots	mBot
5/1/2017 22:01:45	Menen & Ieper	Scratch, Scratch4Arduino, Lego Mindstorms, Arduino, minecraft+javascript / Arduino (via C)	persoonlijke ervaring: minecraft + javascript
5/1/2017 22:16:54	Eeklo	Scratch, Scratch4Arduino, Lego Mindstorms, Arduino, ArduBlocks	Scratch
5/1/2017 22:17:03	Mechelen	Scratch, Scratch4Arduino, Arduino, Lego WeDo, MBots	Scratch
5/1/2017 22:20:53		Scratch, Lego Mindstorms, Arduino, Dwengo	Scratch en Dwengo

Timestamp	Geef voor elke tool uit de vorige vraag aan voor welke leeftijdsgroep u deze tool zou gebruiken?	Welke tips kan u meegeven aan beginnende coaches bij het inzetten van zulke tools om kinderen te leren programmeren?
5/1/2017 20:37:54	7 tot 15	Online veel ideeën te vinden
5/1/2017 20:38:44	Vanaf 8 en ouder indien beginner	Kleine stapjes aanmoedigen
5/1/2017 20:40:30	Scratch: 8-12 jaar - Lego Mindstorms 10-15 jaar	Eerst zorgen dat je het programma zelf onder de knie hebt
5/1/2017 20:41:11	Scratch 7-12, mBots 9-15	Geduld, duidelijk uitleggen en nog meer geduld... KInderen willen wel leren maar je moet ze vrijlaten in wat ze willen leren! Nooit forceren en het initiatief aan de kinderen laten, bijsturen en lessen laten trekken
5/1/2017 20:42:28	9-12	
5/1/2017 20:44:12	7-onbeperkt	Probeer ook zelf van alles uit. Je kan je nooit niet helemaal voorbereiden, want de kids stellen toch vragen die je niet weet, maar als je er zelf wat in geexperimenteerd hebt, gaat het allemaal wat vlotter.
5/1/2017 20:49:06	12+	Unity is geschikt voor kinderen die zoeken naar een uitdaging die verder gaat dan de courante tools die bij Coderdojo worden aangeboden. Ideaal om te differentiëren dus. Wel meer tijdsinvestering vereist van zowel kind als coach.
5/1/2017 21:01:33	8-14j	Goed leermateriaal ter beschikking stellen waarmee ze zelf aan de slag kunnen en dat een snelle 'success-ervaring' geeft.
5/1/2017 21:06:32	7 tot 12 jaar	Laat de kinderen hun fantasie de vrije loop. Zorg dat je die fantasie kan omzetten naar de taal om die fantasie te realiseren.
5/1/2017 21:06:44	8-14	Scratch is de gemakkelijkste tool om te starten. Er zijn ook veel Scratch tutorials (bvb. op YouTube of CS First). Dan kan je met Python (CodeCombat, Pygame, Raspberry Pi enz.) proberen.
5/1/2017 21:27:35	Scratch: tot 12-13 jaar, CodePen: tot +/- 15 jaar	Gebruik de tools zelf ook (voldoende). Vragen van kinderen kunnen ook over het gebruik en/of beschikbare functionaliteiten gaan. Het doel moet steeds primeren op de gebruikte tool.
5/1/2017 21:40:05	leeftijd is niet belangrijk, kennis wel	leg dingen uit maar raak de muis of toetsen bord niet aan
5/1/2017 21:42:14	6-12	Probeer de kinderen alle verschillende functionaliteiten (variabelen, controle-blokken, ...) te gebruiken, niet enkel de voor de hand liggende (uiterlijk, geluid, ...)
5/1/2017 21:44:47	scratch 6 - 12 mbots 9 - 14	zeker zelf me spelen om te leren
5/1/2017 21:46:09	11+, 7+	Basis uitleggen en vervolgens opdrachtblaadjes die specifieke vaardigheden aanleren, dan pas zelf iets volledig creëren.
5/1/2017 22:01:15	Scratch/mbots 6-10	Visueel houden, genoeg doelen bereiken, al spelend leren
5/1/2017 22:01:45	9+	kinderen volgen en ze zoektips geven ipv programeerles
5/1/2017 22:16:54	7 tot 10 jaar	Kinderen initiatief laten nemen. Hun fantasie is eindeloos om met dezelfde "bouwblokjes" een compleet ander resultaat te krijgen.
5/1/2017 22:17:03	Scratch: 6 tem 14, WeDo: 6 tem 10, Mbots: 8 tem 14, Arduino: 10 tem 20	Laat het kind zelf zijn weg zoeken, help, maar doe het niet helemaal voor
5/1/2017 22:20:53	7 - 14 en 9 - 18	Zorg ervoor dat het plezant blijft en niet saai wordt.

Timestamp	In welke CoderDojo(s) bent u actief?	Welke tools worden er in uw CoderDojo gebruikt?	Met welke tool(s) heeft u de beste ervaringen?
5/1/2017 22:32:41	CoderDojo (Hasselt - Diepenbeek)	Scratch, Lego Mindstorms, Minecraft Computercraft/ComputercraftEDU	Lego Mindstorms (Arduino als hobby)
5/1/2017 22:35:20	Zemst	Scratch, Lego Mindstorms, Arduino	Scratch
5/1/2017 22:48:55		Scratch, Scratch4Arduino, Arduino	Scratch
5/1/2017 23:01:58	Hasselt	Scratch, Lego Mindstorms	Scratch
5/1/2017 23:05:02	Gent & De Pinte	Scratch, Lego Mindstorms, Arduino, Greenfoot (Java), Python, HTML, CSS, Javascript	Greenfoot & Java
5/1/2017 23:26:54	Schaarbeek	Scratch, Mbot, Sphereo	Scratch
5/1/2017 23:27:50	Hasselt	Scratch, Lego Mindstorms, Minecraft	
5/1/2017 23:31:44	Hasselt	Scratch, Lego Mindstorms, Minecraft, App Inventor	Scratch
5/2/2017 7:21:17	Edegem	Scratch, Arduino, ArduBlocks, Python, HTML & Javascript	Scratch, Ardublock
5/2/2017 7:25:58	Sint-Truiden	Scratch	Scratch
5/2/2017 8:05:38	Zemst	Scratch, Lego Mindstorms, Arduino	Scratch
5/2/2017 9:09:02	Lier	Scratch, Lego Mindstorms, HTML/CSS, LittleBits, Javascript, Raspberry Pi	Scratch

Timestamp	Geef voor elke tool uit de vorige vraag aan voor welke leeftijdsgroep u deze tool zou gebruiken?	Welke tips kan u meegeven aan beginnende coaches bij het inzetten van zulke tools om kinderen te leren programmeren?
5/1/2017 22:32:41	Lego Mindstorms vanaf 12 en Arduino vanaf 14 maar dit verschilt per jongere	Laat de jongere visueel zien wat elk onderdeel doet. Laat ze zelf iets uit proberen en heb eventueel een paar mini projectjes klaar om te laten zien wat elk onderdeel doet. Leg uit wat de robot (bij lego mindstorms) doet en wat we eigenlijk willen dat hij doet. Zo laat je ze hun eigen denkwijze ontwikkelen.
5/1/2017 22:35:20	Scratch: 6 tot 10 en als inleiding; Arduino: 9 tot 14; Lego Mindstorms: 8 tot 16	Laat de kinderen vooral zelf experimenteren en met vragen komen
5/1/2017 22:48:55	Scratch: daarmee zou ik steeds beginnen, gelijk welke leeftijdsgroep, wanneer ze geen programmeerervaring hebben Scratch4Arduino: diegene die Scratch al kennen en iets tastbaar willen programmeren Arduino: vorige categorie en die nog verder willen gaan.	Leg de zaken uit met begrippen die ze reeds kennen. Bijvoorbeeld een variabele, hoe leg je dat uit? Heb het voorbeeld genomen van het 1ste studiejaar waar ze een doosje hebben met woordjes en een doosje met hun cijfers. Een 'string' variabele is een doosje waarin ze enkel woordjes kunnen steken. Een 'integer' variabele is een doosje waarin ze enkel cijfers kunnen steken. Laat beginners eerst maar wat code 'overtypen' en laat ze dan ontdekken wat er gebeurd als ze een cijfer veranderen bij bvb het aantal stappen dat een figuurtje gaat doen.
5/1/2017 23:01:58	10-15	Laat het kind zijn fantasie werken. Wanneer hij dan een functionaliteit voorstelt, vraag hoe hij er aan zou beginnen. Deel op in kleine deelproblemen die hij zelf kan oplossen.
5/1/2017 23:05:02	11+	begin met dingen die snel resultaat opleveren, zodat de kinderen niet ontmoedigd raken door de complexiteit van code schrijven (Scratch is een stuk simpeler/intuïtiever om mee te starten, maar met Greenfoot/Arduino zijn de mogelijkheden eindeloos veel groter)
5/1/2017 23:26:54	8-14 Jaar	Bouw het materiaal erg traag op
5/1/2017 23:27:50		
5/1/2017 23:31:44	Scratch (jongste), Mindstorms(middelste) en App Inventor (oudste)	Geef een basisuitleg en laat ze zelf zoveel mogelijk ontdekken. Ze zijn slimmer dan je denkt.
5/2/2017 7:21:17	Scratch: va 7j, Arduio/Ardublock: va 9j	Zorg voor een traject / opdrachten zodat het interessant blijft
5/2/2017 7:25:58	10 tot 11	Probeer zelf dingen te maken en hou het simpel
5/2/2017 8:05:38	7-15	Maak de oefeningen zelf eerst thuis
5/2/2017 9:09:02	8-16	Zorg voor opeenvolgende uitdagingen die telkens de mogelijkheden van de tool (Scratch) verder uitdiepen, om zo de kennis bij de deelnemer op te bouwen. De overstap van visuele tools als Scratch naar een meer traditionele programmeeromgeving (Javascript) is niet te onderschatten omdat de deelnemer hier totaal anders mee moet omgaan. Geef die overgang dus voldoende aandacht, en anticipeer op een steile leercurve en wat schijnbare terugval qua kennisniveau omdat de deelnemer niet meer gewoon blokjes kan verslepen.

Timestamp	In welke CoderDojo(s) bent u actief?	Welke tools worden er in uw CoderDojo gebruikt?	Met welke tool(s) heeft u de beste ervaringen?
5/2/2017 9:11:42	Waregem	Scratch, Scratch4Arduino, Arduino, mbot, scratch junior	Scratch & mbot (in combinatie met Scratch)
5/2/2017 9:15:22	Mol	Scratch, Scratch4Arduino, Arduino, mbot, python, codecombat, (www.codecombat.nl)	scratch, arduino, mbot, codecombat
5/2/2017 9:17:48	Rotselaar	Scratch	Scratch
5/2/2017 9:23:21	Torhout	Scratch, App Inventor	Scratch
5/2/2017 9:32:04		Scratch	scratch and makey makey
5/2/2017 9:41:00	Gent	Scratch, Lego Mindstorms, Arduino, Java / Greenfoot	Scratch
5/2/2017 9:55:00	Gent	Scratch, Lego Mindstorms, Arduino, Greenfoot	Scratch & Greenfoot
5/2/2017 9:58:39	Waregem	Scratch, Scratch4Arduino, ScratchJr, mBlock	Scratch & mBlock
5/2/2017 10:20:16	Westerlo	Scratch, Scratch4Arduino, Arduino, Minecraft	Makeblock mBlock
5/2/2017 10:22:53	Zemst	Scratch, Lego Mindstorms, Arduino, HTML	Scratch
5/2/2017 11:54:27	Aalst	Scratch, Lego Mindstorms, Arduino, mbot	Scratch
5/2/2017 12:18:30	Ieper	Scratch, Lego Mindstorms, Arduino, BeeBots, Minecraft	Scratch, Mindstorms

Timestamp	Geef voor elke tool uit de vorige vraag aan voor welke leeftijdsgroep u deze tool zou gebruiken?	Welke tips kan u meegeven aan beginnende coaches bij het inzetten van zulke tools om kinderen te leren programmeren?
5/2/2017 9:11:42	scratch junior (7-9), scratch (10-14), mbot (10-14)	Als het uiteindelijke doel is om kinderen te leren programmeren, zorg er dan voor dat de omgeving waarin je werkt (bv. robots, microcontrollers) niet de aandacht van het programmeren wegtrekt of dat kinderen de illusie krijgen dat programmeren altijd gelinkt is aan een bepaalde omgeving. Hou voor ogen dat voor het programmeren in de praktijk doorgaans gebruikt gemaakt wordt van een tekstuele programmeertaal, en voorzie dus ook een traject dat uitmondt in het werken in een tekstuele programmeertaal. Voor de automatische beoordeling van ingediende programmeeroplossingen zijn we dit jaar aan de UGent beginnen ontwikkelen aan Dodona (dodona.ugent.be). Voorlopig enkel toegankelijk voor UGent-studenten, maar tijdens deze zomer zou het platform voor iedereen toegankelijk moeten worden. We hebben nu reeds judges (die instaan voor de automatische beoordeling) voor de programmeertalen Python, Java, JavaScript, Haskell en Unix/Bash. Er staan nog een paar programmeertalen op de agenda, waaronder ... Scratch !!!
5/2/2017 9:15:22	scratch < 10, mbot > 10, codecombat > 12, arduino > 12	Geef tips, maar laat ze zelf nadenken!
5/2/2017 9:17:48	7-12	Laat kinderen aan interessante problemen / games / projecten werken.
5/2/2017 9:23:21	7-12	stimuleer creativiteit, laat ze ook kijken wat andere kinderen reeds gedaan hebben
5/2/2017 9:32:04	7-16	Moeilijk te zeggen, ben zelf maar sinds een paar weken (lead) coach met pas opgestarte dojo
5/2/2017 9:41:00	10-60	Beef antwoorden over de Tool, stel vragen over de problemen.
5/2/2017 9:55:00	Scratch : 5 - 15 jaar, Greenfoot vanaf 13 jaar	Voorbereiding (zelf eens iets maken thuis), stap voor stap werken, geduldig zijn & blijven.
5/2/2017 9:58:39	Scratch: vanaf 9 jaar ScarchJr: vanaf 6-7 jaar Scratch4Arduino: vanaf 10 jaar mBlock: vanaf 10 jaar	Laat de kinderen zelf ontdekken hoe alles werkt. Begeleiding mag, maar je moet hun handje niet constant vasthouden. Het mag geen school worden, maar een plaats waar ze kunnen experimenteren en nieuwe technologie ontdekken
5/2/2017 10:20:16	Scratch: 7-10 jaar, mBlock: 8-11 jaar, Minecraft: 11-... jaar	loop een eerste keer mee met CoderDojo om te zien hoe de kinderen dit gebruiken, daar leer je het meeste van
5/2/2017 10:22:53	Scratch: 8-12, Lego Mindstorms: 9-16, HTML: 10-16, Arduino: 12-16	
5/2/2017 11:54:27	Tot 12 jaar	
5/2/2017 12:18:30	Scratch vanaf 7, Mindstorms vanaf 10-12	Goede basiskennis en genoeg creativiteit om uitdagingen te bedenken. Bijv. als ze zelf een simpel spelletje gemaakt hebben, ze extra motiveren om variabelen met punten toe te voegen, levels, etc.

Timestamp	In welke CoderDojo(s) bent u actief?	Welke tools worden er in uw CoderDojo gebruikt?	Met welke tool(s) heeft u de beste ervaringen?
5/2/2017 12:39:37	Waregem	Scratch, Scratch4Arduino, Arduino, mBot met mBlock, Scratch Junior, Lightbot.com, hourofcode.com	mBot
5/2/2017 13:09:12	Brugge	Scratch, Lego Mindstorms, Arduino, Python, PHP	Scratch
5/2/2017 13:40:39	Kortrijk	Scratch, Scratch4Arduino, Lego Mindstorms, Arduino, Codecombat.org	Scratch
5/2/2017 16:03:42	Roeselare, Antwerpen	Scratch, Lego Mindstorms, Arduino, Minecraft, Blender	Scratch, Arduino, mindstorms
5/2/2017 20:46:03		Scratch, Lego Mindstorms, Arduino, Sphero, Raspberry Pi	Mindstorms
5/2/2017 21:14:58	SKW	Scratch, Scratch4Arduino, Arduino	scratch
5/2/2017 21:18:31	Klein Brabant	Scratch, Lego Mindstorms, Mbot	Scratch
5/2/2017 23:41:07	Gent	Scratch, Lego Mindstorms, Arduino, Java (Greenfoot)	Scratch, Java
5/3/2017 0:13:14	Waregem	Scratch, Scratch4Arduino, Arduino, mBot	Mbot en arduino

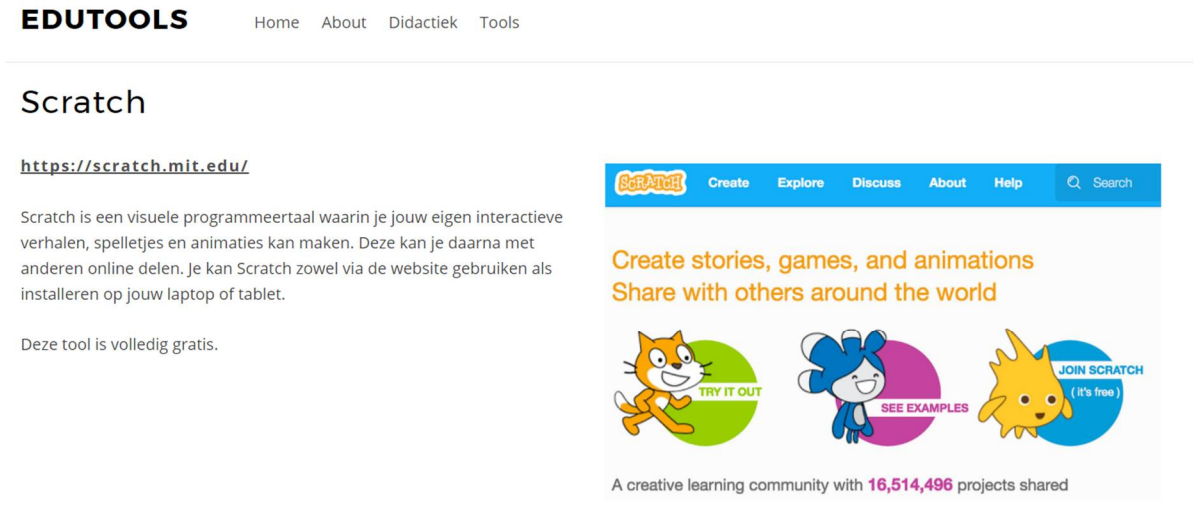
Timestamp	Geef voor elke tool uit de vorige vraag aan voor welke leeftijdsgroep u deze tool zou gebruiken?	Welke tips kan u meegeven aan beginnende coaches bij het inzetten van zulke tools om kinderen te leren programmeren?
5/2/2017 12:39:37	7-8: Scratch Junior, lightbot.com, hourofcode.com, 9-12: Scratch, mBot, 12+ Arduino	In het begin sturend werken en niet vrijblijvend laten werken.
5/2/2017 13:09:12	Scratch: 8 - ..., De rest: 10 -	Vooral niet te veel willen coachen, maar laten ontdekken
5/2/2017 13:40:39	9-12, of voor oudere kinderen die nog nooit code geschreven hebben maar dan slechts voor een korte periode zodat ze de basisbeginselen kunnen leren zoals if-then en for loops	Visualiseer de logica. Als een kind zijn scratch programma niet snapt, laat hem/haar dan opstaan en voer de stappen fysiek uit. Dit helpt vooral bij het uitleggen waarom we herhaallussen nodig hebben bij het detecteren van botsingen tussen sprites.
5/2/2017 16:03:42	7-99, 11-15, 10-14	Maak zelf ook eens wat met de tools dat niet in functie is van coderdojo.
5/2/2017 20:46:03	12 - 17	zorg dat je zelf al veel heb gespeeld met de tools zodat je bij problemen meteen/sneller kan vindenwat het probleem is geef gerichte instructies hou uitdagingen achter de hand voor kinderen met inspiratieblocks
5/2/2017 21:14:58	9-14j	Ken zelf de tool vlot, bekijk voorbeelden online van wat andere maken en probeer de structuur te snappen. Bedenk zelf al wat toffe scenarios. Wees zelf enthousiast. Maak visueel, voeg een of twee blokjes toe en kijk wat er veranderd. Leer kinderen denken waarom iets 'fout' gaat/niet gaat zoals verwacht. Vraag waarom en als ze het niet weten leg uit wat er gebeurd en laat hen zelf denken hoe je dat zou oplossen. Leer kinderen opsplitsen in kleine deeltjes. Ze kunnen niet alles in een keer maken. Waarmee begin je? bv. eerst auw zeggen als je geraakt wordt (detecteren) en dan pas levens verliezen (variabele veranderen). Kleine opdrachten/ideetjes die de kinderen blijven uitdagen op een plaats leggen waar ze allemaal aankunnen. Pas op voor kinderen die altijd hetzelfde maken, maar in een andere vorm, probeer hen iets speciaal te laten doen. Moedig aan om dingen te proberen. Wat zou er gebeuren als we dit aanpassen of dat doen? Moedig ervaren kinderen aan om minder ervaren kinderen te helpen, maar let erop dat ze het niet in hun plaats gaan maken. Toon iets cool dat jezelf hebt gemaakt (of dat je goed snapt) met nieuwe concepten/dingen die niet vaak in je groep gebruikt worden. Bv. Startscherm, highscores, klonen, ...
5/2/2017 21:18:31	7-15	Jezelf openstellen voor het kind, en proberen te denken zoals hij
5/2/2017 23:41:07	Scratch (6-12j) / Lego Mindstorms (vanaf 8j) / Arduino (vanaf 10j) / Java (vanaf 9 j)	Herinner de kinderen dat ze ook eerst gekropen hebben eer ze konden stappen en lopen. Geef ze voldoende zelfvertrouwen door ze complementeren op wat ze wel al gekunnen hebben.
5/3/2017 0:13:14	Scratch: < 10 j en arduino: >10j	Just do it!

Timestamp	In welke CoderDojo(s) bent u actief?	Welke tools worden er in uw CoderDojo gebruikt?	Met welke tool(s) heeft u de beste ervaringen?
5/3/2017 9:53:47	roeselare en kortrijk	Scratch, Scratch4Arduino, Lego Mindstorms, Arduino, App Inventor (android programmeren) + Minecraft	Afhankelijk van ervaring kind: beginneling: Scratch => gevorderden: App Inventor + Scratch4Arduino + Scratch4Minecraft => ervaren: minecraft javascript + arduino c
5/3/2017 10:21:00	westerlo	Scratch, mBots	scr
5/3/2017 16:46:06	Aalter	Scratch, Lego Mindstorms, Arduino	Scratch
5/3/2017 20:59:54	Bierbeek	Scratch, Arduino, Mbots, Mblock, websites (html + css)	Scratch, Arduino
5/5/2017 22:48:06	Mechelen	Scratch, Lego Mindstorms, Arduino, MBot	Geen idee
5/6/2017 10:29:50	Gent	Scratch, Lego Mindstorms, Arduino, Java	Scratch
5/6/2017 10:59:36	Torhout	Scratch, App Inventor	Scratch
5/6/2017 11:40:48	dendermonde, sint niklaas	Scratch, Scratch4Arduino, Lego Mindstorms, Arduino, minecraft, bitsbox, unity, python enz	scratch
5/8/2017 13:28:53	Zulte	Scratch, Lego Mindstorms, Arduino, ArduBlocks	Scratch
5/9/2017 21:31:01	dilbeek	Scratch, Scratch4Arduino, Arduino, appinventor	scratch

Timestamp	Geef voor elke tool uit de vorige vraag aan voor welke leeftijdsgroep u deze tool zou gebruiken?	Welke tips kan u meegeven aan beginnende coaches bij het inzetten van zulke tools om kinderen te leren programmeren?
5/3/2017 9:53:47	cfr vorig antwoord en ik zou eerder ervaringsgroep noemen. De vlaamse Coderdojo community heeft nu app inventor in het nederlands vertaald (officiele release kortelings) waardoor kinderen vanaf 10 jaar apps kunnen maken (voorheen 13 à 14 jaar doordat enkel engels). Hetzelfde gaat op voor Scratch4Minecraft (vertaald door roeselaarse coach friedel verpoort)	Werk op enthousiasme en stimuleer de kinderen. Geef ze een schouderklopke en daag ze uit binnen HUN eigen creatie een stapje hoger te zetten
5/3/2017 10:21:00	scratch 6-11 mbots 10-14	simpele opdrachten/taken die snel feedback geven zijn belangrijk om de kinderen geïnteresseerd te houden. Voldoende opvolgen en prikkelen met steeds lichtjes moeilijkere opdrachten. Probeer een-op-een te werken met de kinderen bij het uitleggen van concepten om hun aandacht vast te houden en de uitleg te personaliseren.
5/3/2017 16:46:06	Scratch voor 8-14j, mindstorm 12+, arduino 12+	Coachen van het kind, niet zozeer leerstof overbrengen. Kind vrijheid geven om eigen interpretatie te geven aan oefening. Sommige zijn liever bezig van grafisch werk, andere met spelelementen,....
5/3/2017 20:59:54	Scratch: vanaf 6-7 jaar tot 12, Arduino vanaf 12	Zorg dat je een aantal simpele voorbeelden/opdrachten hebt waarmee de kinderen meteen aan de slag kunnen (bvb. Scratch cards, Sushi cards...)
5/5/2017 22:48:06	Scratch voor diegene die kunnen lezen, eens ze dit onder de knie hebben kunnen ze een robot aansturen met blokken code, daarna leren ze de code op te pikken en daaruit te leren om uiteindelijk bij Arduino te belanden.	stap voor stap leren programmeren, niet te veel ineens willen
5/6/2017 10:29:50	7-12 jaar	
5/6/2017 10:59:36	Scratch -> vanaf 4e leerjaar Arduino -> vanaf 6e leerjaar of 1e jaar secundair? Met de andere tools heb ik geen ervaring.	Laat hen zo veel mogelijk zelf experimenteren
5/6/2017 11:40:48	7-	
5/8/2017 13:28:53	Scratch: 8 - 16; Lego Mindstorm: 12 - 18; Arduino: 10 - 18; ArduBlocks: 8 - 16	Zelf grondig uitproberen
5/9/2017 21:31:01	all 8-15;only scratch <8	be a coach = support on not tell

7 Website

EDUTOOLS
Home
About
Didactiek
Tools
EDUCATIEVE TOOLS
Wek de interesse van jouw leerlingen
Tools
Kies de tool die
perfect bij jouw doelgroep past.
Didactiek
Krijg heel wat tips voor het vlot
inzetten van deze tools in jouw les.
Lesmateriaal
Deel en maak gebruik van
lesmateriaal via 2link2.


EDUTOOLS
Home
About
Didactiek
Tools
Scratch
https://scratch.mit.edu/
Scratch is een visuele programmeertaal waarin je jouw eigen interactieve verhalen, spelletjes en animaties kan maken. Deze kan je daarna met anderen online delen. Je kan Scratch zowel via de website gebruiken als installeren op jouw laptop of tablet.
Deze tool is volledig gratis.
Create
Explore
Discuss
About
Help
Search
Create stories, games, and animations
Share with others around the world
TRY IT OUT
SEE EXAMPLES
JOIN SCRATCH
(it's free)
A creative learning community with 16,514,496 projects shared

Lego WeDo
https://education.lego.com