



Universiteit
Leiden
The Netherlands

Opleiding Informatica

Computational thinking integreren in de schoolvak Engels:
verhaal vertellen via programmeren in Scratch

Amine Nadif

Supervisors:

Sabiha Yeni & Efthimia Aivaloglou

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

www.liacs.leidenuniv.nl

13/07/2021

Abstract

De toename van digitalisering en informatisering in de huidige maatschappij vraagt om verandering in het Nederlandse curricula. Veel discussie is ontstaan op de vraag welke kennis en vaardigheden van belang zijn om leerlingen voor te bereiden op een snel veranderende maatschappij. Als antwoord hierop is het onderzoek van de SLO tot conclusie gekomen dat "Computational thinking" bij zulke vereiste vaardigheden behoort. Toch blijft de vraag hoe "Computational thinking" moet worden geïntegreerd in het huidige curricula. Momenteel voeren het Radboud Universiteit en Universiteit Leiden een onderzoeksproject als doel om "Computational thinking" te integreren in de huidige curricula. Mijn scriptie focust zich op de integratie van "Computational thinking" in het schoolvak Engels door middel van "Digital Storytelling". De data verkregen van een codasium school wordt geanalyseerd en in deze scriptie uitgebreid besproken. Hiervan zijn er interviews en enquêtes beschikbaar die letten op het toepassen van "Computational thinking", de houding tegenover de lessen, De Pedagogische Content Knowledge (PCK) van de docent en de verkregen kennis van de Engelse grammatica en woordenschat. Uiteindelijk geeft mijn scriptie een beeld over de effectiviteit van de integratie van "Computational thinking" in het schoolvak Engels. Hieruit bleek er een positieve houding te zijn van de scholieren tegenover de lessen, maar kwamen ook obstakels voor zoals de vereiste achtergrond kennis van de docent.

Contents

1	Introductie	1
1.1	Aanleiding	1
1.2	Probleemstelling	1
1.3	Doelstelling	2
2	Theoretische kader	4
2.1	Definitie Computational thinking	4
2.2	Computational thinking integratie	5
2.2.1	Integratie computationele vakken	5
2.2.2	Integratie non-computationele vakken	6
2.3	Digital storytelling	7
2.4	Definitie PCK	7
3	Methode	9
3.1	Onderzoeksopzet	9
3.2	Participanten	10
3.3	Data hulpmiddelen	10
3.3.1	Demographics	11
3.3.2	Practical assignments	12
3.3.3	Learner reports	12
3.3.4	Exit tickets	12
3.3.5	Interviews scholieren	13
3.3.6	Interview docent	13
3.4	Data analyse	14
4	Resultaten	15
4.1	Houdingen van scholieren	15
4.1.1	Scholieren gebruik van technologie	15
4.1.2	Programmeer ervaring van scholieren	17
4.1.3	Zelfvertrouwen van scholieren	18
4.1.4	Geleerde kennis van scholieren	19
4.1.5	Gedrag & moeilijkheden	20
4.2	Computational thinking vaardigheden	23
4.2.1	Algoritmisch denken	23
4.2.2	Generalisatie	24
4.2.3	Evaluatie	24
4.2.4	Decompositie	25
4.3	Pedagogical Content Knowledge	25
4.3.1	M1 : Leerdoelen	26
4.3.2	M2 : Begrip van scholieren	26
4.3.3	M3 : Instructionele strategieën	27
4.3.4	M4 : Evaluatie	28
4.4	Implementatie suggesties van docent	29

5 Conclusie	30
6 Discussie	32
6.1 Reflectie op resultaten	32
6.2 Limitaties	32
6.3 Toekomstige werk	32
References	35
7 Toestemmingsverklaring	36

1 Introductie

1.1 Aanleiding

Er is een toename van technologisering en automatisering over de hele wereld. Dagelijks ervaren we de invloed van digitale technologie en hoe het zich aanpast. Hierdoor is er vraag ontstaan naar programmeervaardigheden in de onderwijssector. Programmeren is een concept dat bestaat uit veel vaardigheden. Om deze vaardigheden te identificeren, zijn er onderzoeken gedaan naar dit concept. De SLO, het nationaal expertisecentrum leerontwikkeling, heeft onderzoek gedaan naar deze vaardigheden met betrekking tot het Nederlandse onderwijs [TFVdH14]. Aanleiding voor het onderzoek was om het begrip “digitale geletterdheid” te verduidelijken naar aanleiding van een rapport van de KNAW [KNA12]. Het KNAW-rapport constateerde hoe de digitale geletterdheid van studenten sterk verbeterd moet worden. De SLO en het ministerie van OCW constateren dat de digitale geletterdheid in het basisonderwijs aandacht behoeft, gezien de steeds digitale wordende wereld waarin we leven. De vraag is echter wat precies bedoeld wordt met digitale geletterdheid, welke aspecten van digitale geletterdheid wenselijk worden geacht voor het onderwijs en hoe het wordt geïmplementeerd in het curriculum. Thijs, Fisser en Van der Hoeven [TFVdH14] analyseerde het KNAW-rapport en vergeleken het rapport met het Nederlandse onderwijs. Ze identificeerden vier hoofdcomponenten van digitale geletterdheid, namelijk IT-basisvaardigheden, mediageletterdheid, informatievaardigheid en computational thinking, waarop de focus werd gelegd bij “Computational thinking”. De auteurs concludeerden ook dat het nauwelijks aanwezig was in de Nederlandse curricula. Een andere analyse van Schmidt [SBB17] concludeerde dat de bestaande curricula voldoende kansen bieden voor de andere hoofdcomponenten zoals informatievaardigheden en mediageletterdheid, maar dat IT-vaardigheden en Computational thinking vaak onvoldoende kansen aangeboden werden. Computational thinking is een essentiële vaardigheid voor digitale geletterdheid en wordt wetenschappelijk gezien als een belangrijke vaardigheid om te bezitten.

Jeanette Wing [Win06] gelooft dat computational thinking een essentieel onderdeel is voor de opvoeding van elk kind. Voogt [VFG+15] is van mening dat er veel meer aandacht nodig is voor manieren om CT-vaardigheden op te nemen in bestaande schoolvakken, zodat leerlingen op creatieve manieren zich kunnen ontwikkelen om computational thinking toe te passen binnen hun respectievelijke disciplines. Gelijkertijd, suggereert Alan Bundy [Bun07] dat het vermogen om computationeel te denken essentieel is voor elke discipline en dat deze computationele denkconcepten al in andere disciplines zijn gebruikt via probleemoplossende processen. Zo blijkt hoe cruciaal computational thinking als vaardigheid is voor de ontwikkeling van studenten. Vele scholen, zowel binnen als buiten Nederland, zijn begonnen met het integreren van computational thinking in hun curriculum. Toch blijft de vraag hoe dit op de meest efficiënte manier kan worden toegepast.

1.2 Probleemstelling

Het is duidelijk hoe belangrijk computational thinking in onze maatschappij is [Win06] en hoe er niet genoeg aandacht voor is in het Nederlandse onderwijs [VFG+15]. Nu blijft de vraag hoe computational thinking geïntegreerd kan worden in het huidige curriculum. Computational thinking is een nieuw concept in de huidige onderwijs waardoor het lastig is om dit aan te leren. Er zijn meerdere onderzoeken gedaan naar manieren om computational thinking te implementeren in

het onderwijs. Meeste onderzoeken richten zich op het vak informatica, om door middel het vak de programmeervaardigheid computational thinking aan te leren. Daarentegen, is het juist van belang om computational thinking te integreren in andere schoolvakken. De integratie van computational thinking in andere schoolvakken helpen de scholieren om daadwerkelijk de vaardigheid te implementeren in andere gebieden buiten het vak informatica. Het is belangrijk dat een individu "computational thinking" kan toepassen buiten het programmeren. Een rapport van de NRC bracht soortgelijke suggesties naar voren, hoe "computational thinking" een cognitieve vaardigheid is die de gemiddelde persoon geacht hoort te bezitten.

Het NRC-rapport[NRC11] toont hoe studenten computational thinking vaardigheden kunnen toepassen in meerdere gebieden, zoals wetenschap en journalistiek. Het rapport laat zien hoe studenten kennis op doen qua computational thinking vaardigheden en het vak waar de integratie plaatsvindt. Hieruit blijkt ook dat het ook belangrijk is hoe docenten computational thinking kunnen implementeren in hun specifieke vak. Docenten moeten de kennis en middelen hebben om computational thinking aan te leren in hun eigen schoolvak. Echter, blijft de vraag hoe docenten het meest effectief en leerzaam de vaardigheid "computational thinking" kunnen toepassen in hun schoolvak.

1.3 Doelstelling

Het is duidelijk dat er focus gelegd moet worden op de bovengenoemde probleem. Zo is er een doorlopend onderzoeksproject [pro] gestart in 2018 over het integreren van computational thinking in schoolvakken binnen het Nederlandse curricula. Het onderzoeksproject betreft de samenwerking tussen het Radboud Universiteit en Universiteit Leiden, gefinancierd door het NRO, National Regieorgaan Onderwijsonderzoek. Het onderzoeksproject doel is het ondersteunen van scholen door middel van praktijkgericht onderzoek, door het ontwikkelen van geschikte computational thinking leerdoelen voor het basis- en voortgezet onderwijs. Het onderzoeksproject streeft ernaar om computational thinking te verbinden met de algemene cognitieve leerdoelen. Het onderzoeksproject heeft al experimenten uitgevoerd in meerdere scholen van basis- en voortgezet onderwijs. Eén van de experimenten, waarvan de data al bekend is, focust op de integratie van computational thinking in het schoolvak Engels. Hierbij moesten de scholieren werken aan een digitale animatie tijdens de CT-geïntegreerde lessen. In de digitale animatie voerden de scholieren een gesprek in het Engels, dit werd gedaan door middel van de programmeertaal "Scratch". De scholieren werden ondervraagd door middel van enquêtes en interviews door de onderzoekers. Hierbij werd er gekeken naar de achtergrond van de scholieren, de ervaringen tijdens de lessen, het proces van hun werking en obstakels die de scholieren tegemoet zijn gekomen. De docent werd ook geïnterviewd over zijn aanpak van lesgeven, zijn perspectief van de lessen en obstakels tijdens de lessen.

Deze scriptie beschrijft de verkregen resultaten van de bovengenoemde enquêtes en interviews. Het doel van de scriptie is om te achterhalen hoe effectief de CT-geïntegreerde lessen zijn. Hierbij wordt er gekeken naar de manier waarop de scholieren CT-vaardigheden opnemen en toepassen. Het is ook cruciaal om een beeld te tonen van de belevenis van de scholieren. Zo kunnen we te weten komen hoe scholieren de lessen ervaren, hoe moeilijk ze het vinden en of ze daadwerkelijk interesse vertonen. Ook wordt de Pedagogische kennis (PCK) van de docent en vormen een beeld van het perspectief van de docent. We willen achterhalen of de docent duidelijk de CT-vaardigheden kan

integreren. Voor beide scholieren en docent willen we ook de obstakels identificeren. Om de bovengenoemde doelstellingen te bereiken, zal deze scriptie de volgende hoofdvraag beantwoorden:

“Hoe effectief is de integratie van computational thinking in het schoolvak Engels door middel van digital storytelling in het Nederlands onderwijs?”

De deelvragen die hiervoor beantwoord zullen worden, zijn:

- Wat zijn de houdingen van studenten ten opzichte van ”Computational thinking” geïntegreerde Engelse lessen?
- Hoe dragen de ”Computational thinking” geïntegreerde lessen bij aan de ”Computational thinking” gerelateerde kennis van studenten?
- Hoe dragen de ”Computational thinking” geïntegreerde lessen bij aan de Engelse kennis van studenten?
- Wat is de Pedagogische Content Knowledge (PCK) van de docent met betrekking tot ”Computational thinking” integratie?

2 Theoretische kader

2.1 Definitie Computational thinking

Momenteel is er geen algemene overeenstemming over de definitie van "computational thinking". Het concept van computationeel denken is bekritiseerd als te vaag, omdat het zelden duidelijk wordt gemaakt hoe het verschilt van andere vormen van denken [TD16]. Er zijn wel meerdere literatuur gepubliceerd over het concept van computational thinking. Hierdoor zijn er meerdere definities van computational thinking ontstaan. "Computational thinking" is niet recent ontstaan, het concept bestond al rond 1950 maar werd beschreven door meerdere vaardigheden, zoals algoritmisch denken en procedureel denken. De term "Computational thinking" werd voor het eerst gebruikt door Seymour Papert in 1980 [Pap80].

Computationeel denken kwam voor het eerst in de computer science educatie gemeenschap in 2006 tevoorschijn in de vorm van een opstel. Het opstel kwam voor in de Communications of the ACM tijdschrift en werd geschreven door Jeanette Wing [Win06]. Het essay suggereerde dat computationeel denken een fundamentele vaardigheid was voor iedereen, niet alleen voor computerwetenschappers, en pleitte voor het belang van het integreren van computationele ideeën in andere vakken op school. Sindsdien zijn er meerdere literatuur gepubliceerd over de definitie van computationeel denken met verschillende perspectieven. Hierdoor is de focus vooral gelegd op het definiëren van computationeel denken. Het is belangrijk om een algemeen geaccepteerde definitie te creëren anders maakt dat onderzoek naar "Computational thinking" ingewikkeld [LM20]. Alhoewel anderen zich geheel focussen op het concept van computationeel denken, zijn er ook die de definitie van computationeel denken willen verduidelijken door integratie. Joke Voogt en Petra Fisser [VFG⁺15] beweren dat het integreren van computationeel denken in het onderwijs helderheid en overeenkomsten bieden in de discussies over computationeel denken. Hierdoor wordt de concepten van Wing vaak gebruikt voor de definitie van computationeel denken wanneer het gaat om de integratie in andere vakken [LSG⁺20]. Daarom wordt de definitie van Wing in deze scriptie ook benut.

De definitie van computational thinking volgens Jeanette Wing:

"Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" [CLW10]

Oftewel *"Computationeel denken zijn het denkprocessen die betrokken zijn bij het formuleren van problemen en hun oplossingen, zodat de oplossingen representatief zijn in een manier waarbij ze effectief uitgevoerd kunnen worden door een informatieverwerkende agent"*

Bij de definitie van computationeel denken behoren kenmerken die het denkproces van computationeel denken definieert. Omdat er geen algemene definitie is van computationeel denken zijn er ook meerdere karakterisering van deze kenmerken. Voor deze scriptie wordt de definiëring van Selby Woollard [SW13] gebruikt omdat de definiëring ook gebruikt wordt in het onderzoek van het NRO [pro].

De kenmerken die Computational thinking definiëren zijn:

- Decompositie : het opbreken van een complex probleem in onderdelen die makkelijker te begrijpen zijn
- Algoritmisch denken : Het creëren van stappen om een probleem op te lossen
- Evaluatie : Het interpreteren en controleren van een oplossing
- Generalisatie : Het hergebruiken van delen van een oplossing voor andere problemen

2.2 Computational thinking integratie

Wanneer computationeel denken op verschillende inhoudsgebieden wordt toegepast, beïnvloedt het de manier waarop leerlingen problemen benaderen en oplossen. Door verschillende manieren te bieden om problemen te benaderen, helpt computationeel denken om succes voor de probleemoplosser te verzekeren [SSSF14]. Door leerlingen de hulpmiddelen en ondersteuning te bieden om nieuwe probleemoplossende methoden te vinden, zal het vertrouwen van leerlingen in hun vermogen om problemen op te lossen worden versterkt. Leraren moeten er voortdurend naar streven om dit gevoel van keuzevrijheid bij hun leerlingen bij te brengen, waardoor hun vermogen om controle te krijgen over hun leven, zowel binnen als buiten de klas en in de toekomst, rechtstreeks wordt beïnvloed [MSCD+10]. Sneider, Stephenson, Schafer en Flick wijzen erop dat wanneer studenten een probleem benaderen met een achtergrond van computationeel denken, hun kennis kan helpen om nieuwe probleemoplossende vaardigheden te ontwikkelen binnen elk inhoudsgebied [SSSF14].

2.2.1 Integratie computationele vakken

De huidige integratie van computationeel denken in het K-12 curriculum komt in twee vormen voor: in computerwetenschappelijke lessen of door het gebruik van computationeel denken technieken in andere vakken. De integratie van computationeel denken in computerwetenschappelijke lessen is wat simpeler dan de integratie van computationeel denken in niet-computerwetenschappelijke lessen. Computationeel denken bestaat uit hoofdconcepten die vereist zijn voor programmeren en andere computer gerelateerde vakken. In de literatuur hebben anderen de focus gelegd op een aparte computerwetenschappelijke vak in plaats van het integreren in andere vakken. Angeli Voogt [AVF+16] zien twee uitdagingen bij de implementatie van een aparte computer wetenschap vak :

- Het ontwerp van het curriculum op basis van een generiek computationeel denkkader
- De kennis die leraren nodig hebben om het curriculum aan te leren.

De eerste issue gaat om de manier van computationeel denken onderwijzen in computerwetenschappelijke vakken. Omdat het concept van computerwetenschap relatief nieuw is vergeleken met andere schoolvakken, zijn er uitdagingen om dit te onderwijzen in basisscholen en in het voortgezet onderwijs. Angeli beschrijft dat er focus gelegd moet worden op problemen uit de echte wereld wanneer er een curriculum wordt gecreëerd voor computerwetenschappelijke vakken. Hierdoor kunnen scholieren daadwerkelijk hun computationeel denken technieken toepassen in andere situaties buiten

school. Voor het tweede probleem gaat het om de voorbereiding van docenten. Het tweede probleem is ook een uitdaging voor de integratie van andere niet-computerwetenschappelijke vakken. Echter, is dit in computerwetenschappelijke vakken lastiger als andere schoolvakken, gezien de voorkennis vereist hiervoor. Computationeel denken is maar één onderdeel van computerwetenschap, bestaande uit meerdere concepten. Andere computerwetenschappelijke concepten kunnen moeilijker overkomen dan het algemene niveau van het curriculum. Armoni [Arm12] beweert dat scholieren tussen de zes en twaalf moeite zullen hebben met het begrijpen van abstractie in de computerwetenschap, een essentieel proces. Hij legt uit hoe abstractie een inherent onderdeel is van informatica dat altijd wordt ingekapseld tijdens het proces van nadenken over en automatiseren van een oplossing voor een probleem. Integendeel, zegt Gibson [Gib12] hoe jonge kinderen abstract kunnen denken wanneer concrete referentiesystemen worden gebruikt om hun denken te situeren.

2.2.2 Integratie non-computationele vakken

In de literatuur zijn er velen onderzoeken en artikelen uitgebracht over de integratie van computationeel denken in andere schoolvakken. We kunnen deze schoolvakken opsplitsen in alfa- en bètawetenschappen. De schoolvakken uit het bèta curriculum zijn wat makkelijker om te integreren gezien de overeenkomsten met het onderwerp computerwetenschap. Waterman [WGP19] beschouwt meerdere niveaus van computationeel denken integratie. Schoolvakken waarbij computationeel denken concepten al in voorkomen zijn makkelijker om daadwerkelijk computationeel denken te integreren. Leraren op het gebied van wetenschap, technologie, techniek en wiskunde (oftewel STEM) die computationeel denken integreren in hun klas kunnen makkelijker probleem oplossende technieken toepassen in hun opdrachten [BHC11]. De STEM-gerelateerde leraren zijn al bekend met computationeel denken concepten, wat de voorbereiding van deze leraren vereenvoudigd. Barcelos [BMV⁺18] beschrijft de overlapping van computationeel denken concepten met wiskundige concepten. Probleem oplossende technieken in de wiskunde overlappen met CT waarbij leraren ook wel eens CT-concepten onderwijzen aan hun scholieren zonder dat ze het weten.

Swaid [Swa15] bemoedigt de integratie van CT in STEM-gerelateerde vakken. Zo worden toekomstige onderzoekers in STEM-disciplines goed opgeleid en voorbereid tegen complexe problemen die zonder CT niet oplosbaar zijn. In Figuur 1 is een weergave te zien van STEM-vakken in universitair niveau, waarbij er CT-elementen worden geïmplementeerd. Als we naar de bovengenoemde concepten van computationeel denken kijken, zien we dat er in elke vak minstens één concept wordt gebruikt. Hierbij zien we hoe scholieren CT-vaardigheden horen te bezitten voor hun toekomst.

Natuurlijk is er ook veel literatuur beschikbaar naar de computationeel denken integratie in alfa schoolvakken. De integratie hiervan is wat ingewikkelder dan de integratie met beta schoolvakken. Met schoolvakken waarbij de focus op geesteswetenschap ligt, zijn er vaak weinig of zelfs geen concepten die overlappen met computationeel denken. Ondanks de uitdaging, brengt CT-integratie in de geesteswetenschap veel positieve veranderingen. Sabitzer [SDMJ18] beschrijft creatieve mogelijkheden om CT te implementeren in taalkundige vakken door middel van modellering. CT-gerelateerde hulpmiddelen zoals modellering kunnen het leren van talen op verschillende manieren ondersteunen en helpen om tekstbegrip te trainen, woordenschat uit te breiden en grammaticaregels visualiseren.

STEM Courses	Computational Thinking Elements						
	Abstraction	Data	Retrieving	Algorithms	Design	Evaluation	Visualization
Biology I*		√	√	√		√	√
Biology II*		√	√	√		√	√
Applied CS*	√	√	√	√	√	√	√
Programming I*	√	√	√	√	√	√	√
Chemistry I*		√	√	√		√	√
Chemistry II*		√	√	√		√	√
College Algebra*	√	√	√				√
Calculus I*	√	√	√				√
Calculus II*	√	√	√				√
Genetics		√	√	√			√
Programming Languages	√	√	√	√	√	√	√
Object-Oriented Programming	√	√	√	√	√	√	√

* Gate-keeping STEM courses

Figure 1: Computacioneel denken elementen voor STEM-schoolvakken, overgenomen van Swaid [Swa15]

2.3 Digital storytelling

Voor het experiment moeten de scholieren een vorm van "Digital storytelling" toepassen in hun project. "Digital storytelling" is een korte vorm van digitale mediaproductie waarmee iemand een aspect van zijn verhaal kan delen. "Digital storytelling" is een relatief nieuwe term die een nieuwe praktijk beschrijft waarbij mensen digitale hulpmiddelen gebruiken om een verhaal te vertellen. "Digital storytelling" heeft verschillende vormen van digitale verhalen, zoals webgebaseerde verhalen, interactieve verhalen, verhalen in de vorm van computerspellen en digitale animaties. De digitale verhalen kunnen worden gebruikt als een middel in de klas om leerstof te integreren met bestaande kennis en vaardigheden uit het curriculum. Studenten kunnen zo individueel of als een team werken om hun eigen digitale verhalen te produceren [Rol06].

Een onderzoek naar "Computational thinking" integratie door middel van "Digital storytelling" toon positieve resultaten. Hierbij gaven de resultaten aan dat de CT-strategie het leren en verhogen van de taal van leerlingen verbetert en hun motivatie. Deze resultaten impliceren het positieve effect van CT-strategie op het versterken van probleemoplossend vermogen van leerlingen die deelnemen aan "Digital storytelling" en verhoogt hun motivatie en prestaties bij het leren van de Engelse taal. [PCWH20]

2.4 Definitie PCK

Pedagogical Content Knowledge, oftewel PCK, is een begrip die voor het eerste vermeld werd door Shulman in 1987 [Shu86]. Shulman beschreef de term als:

"that special amalgam of content and pedagogy that is uniquely the province of teachers, their own special form of professional understanding"

Oftewel *"dat speciale amalgaam van inhoud en pedagogiek dat uniek is voor leraren, hun eigen speciale vorm van professioneel begrip"*

Hierbij vormt hij een verbinding tussen twee belangrijk onderdelen in PCK : vakinhoudelijke kennis en pedagogische kennis. De pedagogische kennis refereert de kennis van de wijze waarop leerlingen kennis opdoen, wijze waarop leermiddelen worden ingezet en de evaluatie en voorbereiding van het leren zelf. De combinatie tussen de twee hoofdbestanddelen vormen een betere inzicht voor docenten om hun leerlingen complexe concepten aan te leren. Van Driel[BLVD08] suggereert hoe PCK het inzicht van docenten in de manieren waarop leerlingen vakinhoudelijk zaken kunnen begrijpen en om een betere begrip te vormen naar de kennis van doceeractiviteiten. Er bestaan meerdere modellen voor het onderzoeken van de PCK van docenten. Voor mijn thesis zal ik gebruik maken van het PCK-model van Magnusson [MKB99]. Het model van Magnusson is afgeleid uit oudere modellen van Shulman en Grossman[Gro90]. Ondanks het model uit 5 componenten bestaat, benadrukken Barendsen Henze[BH19] het belang van vier van de vijf componenten gebaseerd op hun simpliciteit, pedagogische volledigheid en de betrekking tot de vakinhoud van het onderwijs. Hierbij bestaan de componenten, wat we verwijzen naar M1 tot M4 in volgorde, uit :

M1 : Kennis over doelen en doelstellingen voor het onderwijzen van een specifiek onderwerp in het curriculum.

M2 : Begrip van scholieren over het onderwerp.

M3: Instructiestrategieën met betrekking tot het onderwerp.

M4: Manieren om het begrip van scholieren over het onderwerp te beoordelen.

Voor M1 kijken we naar de leerdoelen bij het onderwijzen van de docent. Hierbij willen we achterhalen wat de docent de scholieren wilt aan leren met zijn nieuwe manier van lesgeven.

Voor M2 kijken we naar het perspectief van de leraar, hoe hij kon bepalen of scholieren daadwerkelijk de lessen hebben begrepen en hoe ze het hebben ervaren.

Voor M3 wordt er gekeken naar de strategieën die worden toegepast om de scholieren de lesstof aan te leren. Er wordt hier ook kritisch gekeken naar de strategieën gebaseerd op het schoolvak Engels en de computationeel denkende vaardigheden.

Voor M4 wordt de evaluatie van de docent geïdentificeerd. Hierbij willen we achterhalen hoe de docent terug kijkt naar de CT-geïntegreerde lessen.

3 Methode

3.1 Onderzoeksopzet

Voor mijn onderzoeksmethode voer ik mijn onderzoek uit als een case-study. Yin [Yin84] definieert de onderzoeksmethode voor case-study's *“als een empirisch onderzoek dat een hedendaags fenomeen onderzoekt binnen zijn reële context; wanneer de grenzen tussen fenomeen en context niet duidelijk zichtbaar zijn; en waarin meerdere bronnen van bewijs worden gebruikt”*.

Een case-study is een kwalitatief onderzoek waarin verschillende methodieken worden ingezet om een sociaal fenomeen te bestuderen. Binnen een case-study wordt er gebruik gemaakt van verschillende informatiebronnen. Er zijn meerdere methodes die in een case-study kunnen worden toegepast om zulke informatiebronnen te bemachtigen, waaronder participatieve observatie, documentenonderzoek en diepte-interview. case-study-methode stelt een onderzoeker in staat om de gegevens binnen een specifieke context nauwkeurig te onderzoeken. In de meeste gevallen selecteert een case-study methode een klein geografisch gebied of een zeer beperkt aantal individuen als onderwerp van studie. case-study's verkennen en onderzoeken hedendaagse sociale fenomenen door middel van een contextuele analyse van een aantal gebeurtenissen of condities, en hun relaties.

Hierbij bestaan er meerdere categorieën voor een case-study, waarvan deze case-study wordt beschouwd als een instrumentele case-study. Een instrumentele case-study selecteert de onderzoeker een kleine groep proefpersonen om een bepaald kenmerk of patroon te onderzoeken, in deze geval om de effectiviteit van de integratie van computationeel denken te bestuderen.

Het doel van de thesis is om een duidelijk beeld te vormen over de effectiviteit in de integratie van computational thinking in het schoolvak Engels. Het is belangrijk dat de scholieren daadwerkelijk computationeel denken concepten kunnen toepassen in het Engels schoolvak. Omdat computationeel denken vooral wordt gebruikt bij het programmeren, kunnen er obstakels vormen bij de integratie ervan. Hierbij wordt er ook gekeken naar de kennis van de docent. In het Nederlands onderwijs, zullen meeste docenten niet getraind zijn in het coderen of andere ICT-gerelateerde vaardigheden, waaronder computationeel denken. Het is ook van belang om het gedrag, de ervaring en denkproces van de scholieren te analyseren om een duidelijk beeld te vormen over de manier van computationeel denken implementatie van de leerlingen.

Mijn thesis maakt deel uit van een onderzoek gebaseerd op de integratie van computationeel denken in het onderwijs. Het doel van het onderzoek is om binnen het onderwijs computationeel denken te associëren met algemene cognitieve leerdoelen door middel van integratie in de drie verschillende leergroepen, waaronder STEM-gerelateerde vakken, taal en geesteswetenschappen. Dit wordt gedaan door middel van praktijkgericht onderzoek, waarbij geschikte computationeel denken leerdoelen worden ontwikkeld om onderwijs- en leerstrategieën van deze doelstellingen te bevorderen [pro]. In het onderzoek worden er experimenten met meerdere scholen uitgevoerd. Mijn thesis maakt deel uit van één van deze scholen en de data in deze thesis is afkomstig van de resultaten binnen de gerelateerde experiment van het onderzoek. In het experiment moeten de scholieren werken aan een project waarbij ze een digitale verhaal moeten maken in scratch. Het digitale verhaal werd door middel van storyboards ontwikkeld en de scholieren moesten hun eigen

stem inspreken waarbij ze Engels spraken. De scholieren werkten aan het project in groepen van twee.

Hierbij zijn er doelen gesteld wat de leerlingen aan het eind van het experiment horen te beheersen. Dit zijn doelen op het gebied van computationeel denken en het Engelse taal. De doelen zijn:

- Leerlingen zijn in staat om in de tegenwoordige tijd vragen over zichzelf te stellen en beantwoorden
- Leerlingen kunnen vragen stellen en beantwoorden over hobby's
- Leerlingen kunnen praten over de datum en tijd
- Leerlingen kunnen plaatsen en andere dingen beschrijven
- Leerlingen kunnen een digitaal verhaal creëren in de programmeertaal Scratch
- Leerlingen hun aangeleerde grammatica en woordenschat toepassen in hun digitale verhalen

Het experiment werd gesplitst in vier lessen. In de eerste les werd het proces van het project uitgelegd aan de scholieren en werd er een planning gemaakt. De scholieren hebben al in het schoolvak Engels geleerd om de grammatica en woordenschat toe te passen in hun digitale verhalen. Ze hebben ook geleerd hoe je vragen over jezelf kan stellen en beantwoorden in de tegenwoordige tijd. De onderwerpen rondom de verhalen gingen om kleding, hobby's, eten, drinken en hun woonplaats. In de tweede les moesten de scholieren hun script uitschrijven en kregen feedback van de docent. In de daaropvolgende twee lessen werden studenten gevraagd om hun eigen digitale verhalen te schrijven. In het verhaal stelden ze zichzelf voor en beschreven hun school, stad en activiteiten buiten school. De scholieren werken in tweetallen en werken aan hun project gebaseerd op het script die ze in de tweede les moesten creëren. Vervolgens schreven de scholieren hun digitale verhalen in de programmeertaal Scratch. De scholieren namen hun dialoog op en voegden ze toe aan hun digitale verhaal. De scholieren werden becijferd op 4 onderdelen waaronder hun inhoud, hun grammatica woordenschat, hun creativiteit en het uitspreken van hun Engelse tekst.

3.2 Participanten

De scholieren behoren allemaal in een technasium school en zitten in de eerste klas van het Voortgezet onderwijs/Hoger algemeen voortgezet onderwijs. De school wordt ook beschouwd als een codasium school. Codasium is een nieuw onderwijsstroom waarbij coderen centraal staat. Buiten coderen, wordt er ook aandacht besteed aan de invloed van digitale technologie in het alledaagse leven. Deze vorm van educatie waarop de focus wordt gelegd op digitale vaardigheden is pas ontstaan en momenteel wordt er samengewerkt met meerdere organisaties om van coderen een eindexamenvak te maken [Cod].

De docent die het schoolvak "Engels" lesgeeft heeft elf jaar ervaring als leraar. De docent heeft geen programmeer ervaring. De docent is een man. De scholieren zijn gesplitst tussen drie klassen. Met alle klassen gecombineerd, zijn er 44 participanten in totaal. De participanten zijn tussen de elf en veertien jaar oud.

3.3 Data hulpmiddelen

Het onderzoek heeft op twee verschillende manieren data verzameld. Door middel van interviews en enquêtes. Mijn scriptie maakt gebruik van transcripties van interviews en de beantwoorde enquêtes

van de scholieren.

Aan het eind van het project hielden de onderzoekers een interview met de leerlingen. Tijdens het interview moesten de scholieren hun project tonen aan de interviewer. De interviewer vraagt vervolgens over het project, de denkwijze van de leerlingen tijdens het maken van het project, moeilijkheden die de leerlingen overkwamen, kennis over Scratch, hun ervaringen en aan het eind werd er gesproken over de andere animaties die Scratch te bieden had. Ook werd de docent geïnterviewd over zijn werkwijze en obstakels die hij tegenkwam tijdens de lessen. Dit werd allemaal opgenomen voor analyse over de CT-geïntegreerde Engelse lessen.

De leerlingen moesten voor, tijdens en na het project enquêtes invullen. In Tabel 1 kun je de exacte week zien waarop de scholieren de enquêtes hadden ingevuld. De enquêtes bevatten verschillende soorten vragen over het project, hun geschiedenis met technologie/ICT, het proces tijdens het project en het eindresultaat. De interviews werden in de laatste week uitgevoerd.

De ouders van de scholieren en docent hebben een toestemmingsformulier ingevuld, waarin zij toestemming geven voor de interviews en dat de participanten vrijwillig deelnamen aan het onderzoek. Zie sectie 7.

Hulpmiddelen	1e week	2e week	3e week	4e week
Toestemmingsformulieren	X			
Demographics	X			
Exit tickets		X	X	
Practical Assignments		X	X	
Learner reports			X	
Interview Docent				X
Interview Scholieren				X

Table 1: Tijdstippen van de formuliers en interviews

3.3.1 Demographics

De eerste enquête die de scholieren moesten invullen is de "Demographic". Dit werd gedaan tijdens de eerste les wanneer het project werd geïntroduceerd aan de scholieren. De focus van de eerste enquête werd gelegd op de kennis over ICT en programmeren. We willen weten wat voor IT-vaardigheden de scholieren al beheersen en hoe ervaren de scholieren zijn op het gebied van programmeren. Ook kijken we naar het zelfvertrouwen van de scholieren wanneer het gaat om programmeren. Verder willen we de algemene gegevens ook weten zoals leeftijd en geslacht. De enquête bestaat uit open en gesloten vragen.

De scholieren moesten invullen of ze eerder een computer, tablet of smartphone hebben gebruikt in de vorm van een meerkeuze vraag. Er werd vervolgens gevraagd naar de ervaring van hun gebruik met computers en waarvoor ze allemaal technologie gebruiken in de vorm van meerkeuze vragen. Aan het eind van de enquête wordt er meer focus gelegd op de programmeervervaring van de scholieren waarbij er gevraagd wordt naar de scholier's geschiedenis met programmeren en welke programmeertalen de scholier kent, waarvan de laatstgenoemde als een mix tussen een meerkeuze en openvraag werd gesteld.

3.3.2 Practical assignments

De volgende enquête is de "Practical assignment". De enquête bevat meerdere vragen bedoeld tijdens andere periodes tijdens de CT-geïntegreerde lessen. De focus van de enquête werd gelegd op het project zelf en de denkwijze van het hele proces. Hierbij willen we erachter komen hoe de scholieren ten werk gaan tijdens het experiment. Het is belangrijk om te zien of ze bepaalde vaardigheden, waaronder computationeel denken vaardigheden, zullen toepassen bij de planning. Alle vragen in de enquête zijn openvragen.

De eerste periode is voor het ontwerp van de animatie. Hierbij moesten de scholieren invullen waar het project over ging en wat de scholieren al wisten over het onderwerp van het project. Als de scholieren voldoende over het onderwerp wisten, konden de scholieren de volgende vraag beantwoorden waarbij er gevraagd werd voor een algoritme gebaseerd op hun onderwerp. Het algoritme moest het onderwerp beschrijven in de vorm van een als-dan redentatie en mochten alleen kernwoorden gebruiken. De manier van schrijven van het algoritme is soortgelijks als de manier van programmeren als het gaat om de bekende "If..Else.." condities wat veel terugkomt in meeste programmeertalen. Verder, wanneer de scholieren al hebben geprogrammeerd, werd er gevraagd naar conflicten waar de scholieren mee vast kwamen te zitten en hoe ze zulke conflicten oplossen als die er waren. Wanneer de scholieren klaar zijn met het project, konden ze de laatste vragen van de enquête invullen. Hierbij werd ze gevraagd over het eindresultaat en hun ervaringen van de lessen.

3.3.3 Learner reports

De volgende enquête werd aan het eind van het experiment ingevuld door de scholieren. De enquête die werd ingevuld is het "Learner report". Bij de enquête werd er gevraagd naar de kennis die de scholieren hebben verkregen tijdens het onderzoek. Hierbij willen we weten of de scholieren wat nieuwe kennis hebben binnengekregen na afloop van de CT-geïntegreerde lessen. We willen zien of de scholieren eigenschappen van computationeel denken vaardigheden zullen opschrijven bij de antwoorden en hoe de scholieren programmeren ervaren na afloop van het experiment.

De scholieren moesten opschrijven wat zij allemaal hebben geleerd op het gebied van regels, uitzonderingen en over de lessen in het algemeen. Dit werd als een openvraag gesteld, waarbij de scholieren een incompleet zin moesten afmaken. De zin had de standaard vorm "Ik heb geleerd dat ...". Hierbij moesten de scholieren opschrijven wat ze niet moeten doen bij een project in de standaard vorm "Ik heb geleerd dat ik niet ...".

3.3.4 Exit tickets

De "Exit tickets" bevatten vragen over de mening van de lessen in de vorm van meerkeuze- en openvragen. De exit tickets werden ingevuld tijdens de tweede en derde week. Hierbij willen we erachter komen hoe de scholieren het hele experiment ervaren. Het is ook belangrijk om te weten of de scholieren de lessen begrijpen en hoe moeilijk de scholieren het hele project vinden.

De enquête bevat bepaalde uitspraken over de les, zoals "Ik heb genoten van de les" waarbij de scholieren 3 opties konden selecteren waaronder : "Ja", "Nee" en "Onzeker". Er werd ook

gevraagd of de scholieren de lessen begrepen, of ze het moeilijk vonden en of ze de les interessant vonden. Ook konden de scholieren commentaar opschrijven over de les zelf.

3.3.5 Interviews scholieren

Deze thesis maakt gebruik van transcripties van interviews. De interviews vonden plaats in de school waarbij de scholieren werden geïnterviewed als een team van twee of één scholier die het team representeert. Hiervan zijn er 7 verschillende interviews gedocumenteerd voor analyse die ook worden besproken in deze thesis. In de interviews moesten de scholieren een gesprek voeren met één onderzoeker. In het interview moesten de scholieren hun eindproject laten zien aan de onderzoeker. Vervolgens worden er vragen gesteld over het project zelf. De interviews hebben vaste vragen die bij elke interview werd gesteld, en vragen die werd gesteld gebaseerd aan de situatie. De scholieren konden in het Engels of in het Nederlands de vragen beantwoorden.

De onderzoeker vraagt aan het begin voor een beschrijving van het project voordat ze samen het project gaan bekijken. Nadat de onderzoeker het project heeft gezien, begint de onderzoeker vragen te stellen over het project zelf en het proces van het maken van het project. Vervolgens, worden er vragen gesteld over het programmeren en de programmeertaal Scratch, de programmeertaal waarin de scholieren het project in hebben gemaakt. Daarna vraagt de interviewer over problemen waaraan de scholieren tegen aan kwamen en hoe ze het hebben opgelost. Hierbij let de onderzoeker vooral op computationaal thinking vaardigheden die zijn toegepast door de scholieren. Verder werd er gevraagd naar de meningen en ervaringen over het project en de programmeertaal Scratch. Aan het eind, toont de onderzoeker andere voorbeeld projecten die in Scratch werden gemaakt en vraagt naar de opmerkingen en meningen van de scholieren.

De bovengenoemde paragraaf beschrijft de algemene structuur van de interviews, maar elke interview bevat bepaalde fragmenten waarbij de onderzoeker zich focust op punten die handig kunnen zijn voor het onderzoek, voornamelijk computationeel denken aspecten en andere interessante onderwerpen voor het onderzoek.

3.3.6 Interview docent

De docent werd geïnterviewed door een onderzoeker. Het interview heeft digitaal plaatsgevonden. Het doel hiervan is om de PCK gerelateerd tot computationeel denken te achterhalen van de docent. Hierbij willen we meer weten over de kennis van de wijze waarop leerlingen geven, de wijze waarop leermiddelen kunnen worden ingezet bij het leren, het klassenmanagement en de lesvoorbereiding en -uitvoering [VFG⁺15]. Het is belangrijk om een goed beeld te krijgen van het perspectief van een docent om te zien wat voor obstakels docenten kunnen tegenkomen en hoe zij zelf het experiment ervaren.

Bij het interview met de docent werden de lessen en het experiment besproken. De onderzoeker vraagt aan het begin over de leerdoelen van het project. Hierbij wordt er naar de Engelse en computationeel denkende aspecten van de leerdoelen gevraagd. Er wordt gevraagd naar de standaard manier van lesgeven en naar de integratie van computationeel denken op het gebied van instructie strategieën. Daarna werd er gevraagd naar de reacties en ervaringen van de scholieren

vanuit het perspectief van een docent. Vervolgens legt de onderzoeker weer de focus op leerdoelen en computationeel denken. De onderzoeker vraagt naar de evaluatie van de leerdoelen, de verschillen en overeenkomsten tussen de standaard manier van lesgeven en de aangepaste manier tijdens het onderzoek. Aan het eind van de interview, wordt er gevraagd naar problemen en verbeteringen binnen het project, en de mening van de docent over het experiment in het algemeen.

3.4 Data analyse

Voor de analyse worden twee manieren van werkwijzen gebruikt, een descriptieve werkwijze en een inductieve werkwijze. De demographics en exit tickets worden op een descriptieve manier geanalyseerd terwijl de interviews, practical assignment en learner reports op een inductieve manier worden geanalyseerd.

Voor de descriptieve analyse wordt er gebruik gemaakt van kwantitatief data. De verkregen data wordt opgeschoond en door middel van grafiek en tabellen in de “Resultaten” sectie weergegeven. Hierbij gebruiken we maten van frequentie.

Voor de inductieve analyse wordt er gebruik gemaakt van kwalitatieve data. Het proces wordt hierbij gesplitst tussen drie fasen, namelijk open-, axiaal- en selectief coderen. De drie fasen worden toegepast in alle data. Tijdens de eerste fase, worden stukjes gegevens vanuit de data gelabeld. Dit wordt gedaan met relevante fragmenten die we samenvatten gebaseerd op de betekenis die wordt weergegeven in de data zelf, dus niet uit bestaande theorie. De labels, oftewel code, krijgen bepaalde eigenschappen die zullen helpen met verbindingen maken naar andere codes. Daarna beginnen we met de tweede fase, axiaal coderen. Hierbij proberen we de relaties tussen de codes van de eerste fasen te identificeren. In de derde fase, selectief coderen, zoeken we naar een kernvariabele die alle gegevens bevatten om zo tot een conclusie te komen en een antwoord te vinden naar de onderzoeksvragen [SC90]. Voor elke set van data worden andere codes toegepast, soms worden er soortgelijke codes toegepast in andere datasets.

Voor de data analyse wordt de computer programma “Atlas.ti” gebruikt. Atlas.ti is een programma die vooral wordt gebruikt, maar niet exclusief, voor kwalitatief onderzoek of kwalitatief data analyse. Het doel van ATLAS.ti is om onderzoekers te helpen bij het ontdekken en systematisch analyseren van complexe verschijnselen die verborgen zijn in ongestructureerde gegevens. Het programma biedt tools waarmee de gebruiker bevindingen in primair gegevensmateriaal kan lokaliseren, coderen en annoteren, om het belang ervan af te wegen en te evalueren, en om de vaak complexe relaties daartussen te visualiseren [LS07].

4 Resultaten

4.1 Houdingen van scholieren

In deze sectie kijken we naar de ervaringen en meningen van de scholieren. We willen weten de scholieren de CT-geïntegreerde lessen ervaren. Er worden vragen gesteld op het gebied van het schoolvak Engels en "computational thinking" vaardigheden waarbij we willen achterhalen wat ze hebben geleerd en of ze het hebben begrepen. Ook wordt er gekeken naar de moeilijkheden en obstakels die voor kwamen voor de scholieren.

In Tabel 2 zien we de leeftijd en geslacht van de scholieren. Bepaalde participanten hebben vragen deels of niet ingevuld wat ook aangegeven wordt met "Niet gespecificeerd".

Leeftijd	Frequency (n)
Elf	4
Twaalf	31
Dertien	6
Veertien	1
Niet gespecificeerd	2

Geslacht	Frequency (n)
Jongen	24
Meisje	18
Anders	1
Niet gespecificeerd	1

Table 2: Leeftijd en geslacht van participanten

4.1.1 Scholieren gebruik van technologie

De participanten werden gevraagd een enquête in te vullen voordat ze aan het project begonnen. De enquête bevat meerkeuze en open vragen over de geschiedenis met ICT en technologie. Er werd ook gevraagd naar de programmeerervaring van de scholieren. In de eerste vraag van de enquête werd de vraag gesteld of de participanten gebruik hebben gemaakt van een computer, tablet en/of smartphone. Dit is in de vorm van een meerkeuzevraag gesteld waarbij de scholieren konden kiezen tussen een "Ja" en een "Nee". Alle participanten hebben minstens één of meer geselecteerd. Geen scholier heeft aangegeven dat die nooit met de bovengenoemde elektronica gebruik van heeft gemaakt of zelf bezit.

In Grafiek 2 zien we resultaten van de participanten op de vraag "Met welke programmeertalen heb je geprogrammeerd?". De vraagstelling is gemixt tussen een open vraag en meerkeuzevraag. De scholieren zagen voorbeeld programmeertalen die ze konden selecteren, maar hadden ook de keuze om zelf programmeertalen in te vullen. Bij de vraag konden scholieren meerdere programmeertalen opschrijven waarmee ze eerder hebben geprogrammeerd. De enige optie waar dat niet zo is, is het antwoord "niet gespecificeerd", waarbij de scholieren heeft aangegeven dat die geen van de programmeertalen kent of waarbij de scholieren niks hebben opgeschreven.

De programmeertaal waarmee veel scholieren al eerder in hebben geprogrammeerd is Python. Python is een objectgeoriënteerde programmeertaal die bekend staat om zijn code leesbaarheid. De taalconstructies en objectgeoriënteerde benaderingen in Python helpen programmeurs om

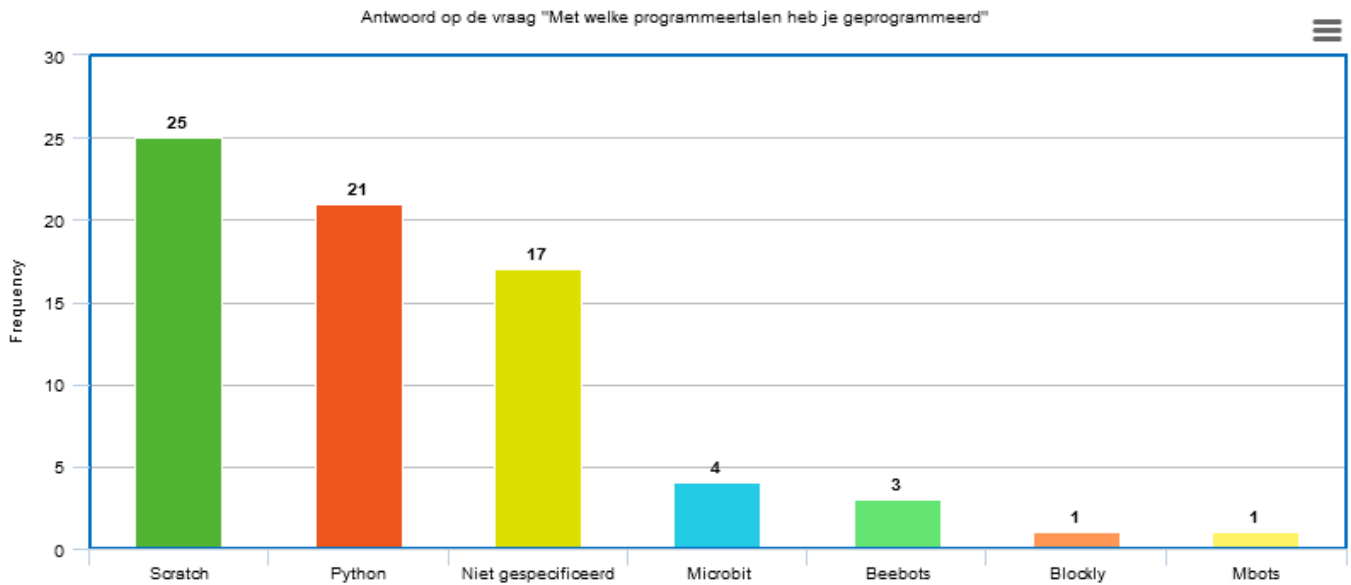


Figure 2: Aantal antwoorden op de vraag "Met welke programmeertalen heb jij geprogrammeerd?"

duidelijke, logische code te schrijven voor klein en grootschalige projecten. Samen met Python, is de programmeertaal Scratch ook vaak gebruikt door de participanten. Scratch, net als Python, is een objectgeoriënteerde visuele programmeertaal die voornamelijk bedoeld is voor kinderen tussen de acht en zestien jaar oud. Scratch is vooral geschikt voor het maken van visualisaties zoals spellen en animaties. Wat Scratch anders maakt vergeleken met de andere programmeertalen is het feit dat Scratch gebruik maakt van blokje in plaats van tekst. Hierdoor worden veel syntax fouten beperkt door de onderdelen van het programma net als legosteentjes aan elkaar te laten passen. Scratch is ook de programmeertaal die gebruikt wordt voor het project.

Verder heeft een minderheid al eerder geprogrammeerd met de programmeertalen Microbit, Beebots, Blockly en Mbots. Dit zijn programmeertalen waarbij de meeste gebruikers jongeren of nieuwelingen in ICT zijn. Verder hebben 21 van de 44 participanten niet eerder geprogrammeerd met een programmeertaal of het niet doorgegeven. Dit wordt aangegeven met "Niet gespecificeerd".

In Tabel 3 zijn de resultaten te zien waarbij er werd gevraagd waarvoor scholieren technologie gebruiken. De scholieren konden kiezen tussen vier antwoorden die te zien zijn in Table 3. Hierbij willen we achterhalen waarbij de scholieren technologie gebruiken en hoe vaak de scholieren er gebruik van maken.

"Ik gebruik technologie voor:"	Antwoord "Nooit"	Antwoord "Soms"	Antwoord "Zelden"	Antwoord "Vaak"	Totaal
"Schrijven van teksten"	1	2	16	26	44
"Maken van tekeningen en dergelijken"	16	8	17	3	44
"Onderzoek"	0	8	2	33	43
"Maken en presenteren van presentaties"	1	10	5	28	44
"Communicatie met anderen"	2	4	15	23	44
"Schrijven van blogs of bijdragen aan wiki's"	22	14	4	4	44
"Gebruik van social networking websites"	1	9	6	26	42
"Maken van muziek, filmpjes of webcasts"	15	13	7	7	42

Table 3: Resultaten op de vraag "Waar gebruik jij technologie voor?"

4.1.2 Programmeer ervaring van scholieren

In Tabel 4 staan de resultaten van de participanten op de vraag "Heb je ooit programmeerlessen gehad?". De vraagstelling is in de vorm van een meerkeuze vraag opgesteld waarbij je drie opties kan aanvinken: "Programmeerlessen gehad", "Geen programmeerlessen gehad" en "Zelfstudie". De tabel laat zien dat een meerderheid wel programmeerlessen in het verleden hebben gehad. De school biedt zelf codeerlessen aan de scholieren als een keuzevak. Van de 44 participanten hebben maar 4 scholieren geen vorm van programmeerlessen gehad. Er zijn ook 4 participanten die aangeven dat ze zichzelf programmeren hebben aangeleerd. Een grotendeel van de participanten hebben al ervaring en kennis over programmeren. Dit was ook verwacht sindsdien de school zelf programmeerlessen aanbiedt vanaf de eerste klas.

Programmeerlessen gehad	Frequency (n)
Ja	36
Nee	4
Nee, programmeren zelf geleerd	4
Totaal	44

Table 4: Resultaten op de vraag "Heb je ooit programmeerlessen gehad?"

Antwoorden	Frequency
Niet gespecificeerd	2
Minder dan 1 maand	7
1 maand tot 1 jaar	27
2 tot 3 jaar	4
Langer dan 3 jaar	0
Totaal	40

Table 5: Resultaten op de vraag "Hoelang heb je programmeerlessen gehad?"

In Tabel 5 zien we de resultaten van de 40 participanten die "programmeerlessen gehad" of "Zelfstudie" hebben aangevinkt, wanneer ze werden gevraagd voor hoelang ze zichzelf hebben geëduceerd over programmeren en soortgelijke vaardigheden. Een grotendeel van de scholieren hebben minstens meer dan 1 maand programmeerlessen of zich zelf geëduceerd op het gebied van programmeren. De twee scholieren die niks hebben ingevuld (valt onder "Niet gespecificeerd"), behoren tot de groep die ook "Zelfstudie" hebben aangevinkt.

4.1.3 Zelfvertrouwen van scholieren

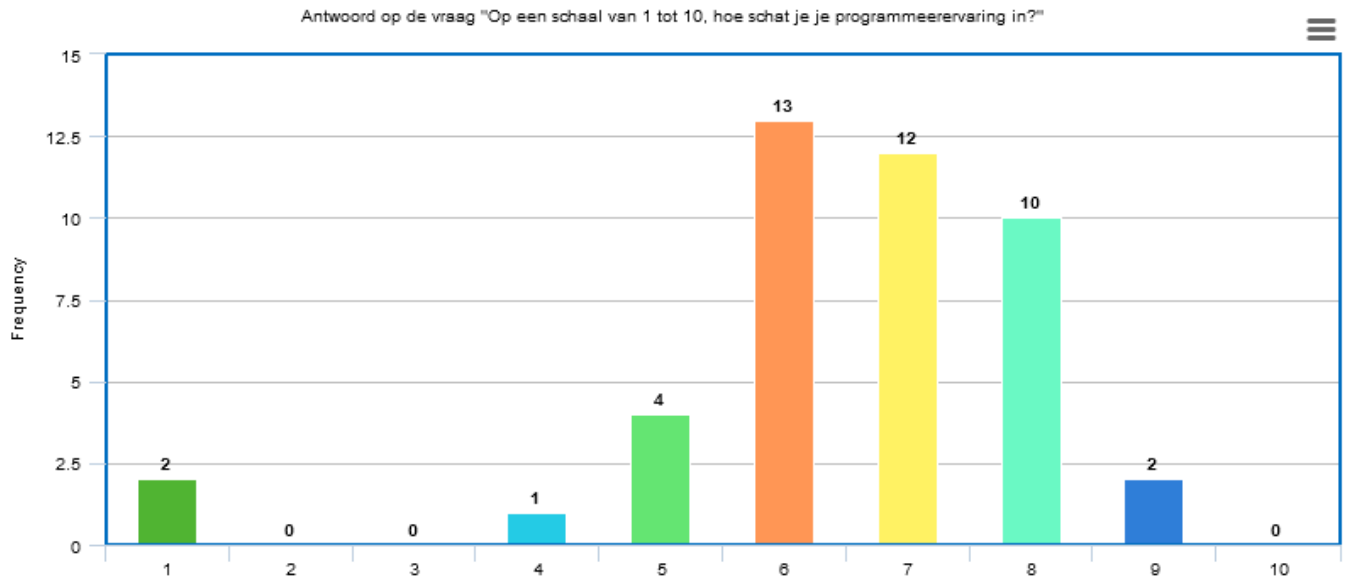


Figure 3: Alle antwoorden op de vraag "Op een schaal van 1 tot 10, hoe schat je je programmeerervaring in?"

In Figuur 3 staan de resultaten van de participanten op de vraag "Hoe schat jij je programmeerervaring in?". De vraagstelling is in de vorm van een meerkeuze vraag waarbij je een antwoord kan geven gebaseerd op een schaal van 10. Aan de grafiek te zien beschouwt de meerderheid van de scholieren een voldoende of goed begrip te hebben over het programmeren. Een minderheid vinden

dat ze een onvoldoende of slecht kennis hebben over het programmeren. Gemiddeld hebben de scholieren een goed zelfvertrouwen over hun IT-vaardigheden. Naast de scholieren, heeft de docent geen specifieke training gekregen.

Vragen	Frequency
"Ik ben een professioneel"	1
"Ik voel me op mijn gemak tijdens het programmeren"	21
"Ik heb een beetje hulp nodig bij het programmeren"	18
"Ik heb veel hulp nodig bij het programmeren"	2
Totaal	42

Table 6: Resultaten voor de vraag "In hoeverre voel je je op je gemak als je aan het programmeren bent?"

In Tabel 6 zien we de resultaten van 42 scholieren over hun gemak met programmeren. De vraagstelling was gedaan door middel van een meerkeuze vraag waarbij de scholieren 4 opties konden kiezen, zoals te zien in Figuur 6. Van de 44 participanten hebben 2 niks ingevuld. Het is belangrijk om te weten hoeveel zelfvertrouwen de scholieren hebben in hun programmeervaardigheden. De meerderheid van de scholieren hebben gekozen tussen "op mijn gemak tijdens het programmeren" en "een beetje hulp bij het programmeren". Van de 2 scholieren die hebben aangegeven dat ze "veel hulp nodig" hebben bij het programmeren, hebben ze beide aangegeven dat ze wel programmeerlessen hebben gehad van minstens één maand. Dit zijn ook dezelfde scholieren die een "1" hebben beantwoord bij de vraag "Hoe schat jij je programmeerervaring in op een schaal van 1 tot 10?". Zie Figuur 3.

4.1.4 Geleerde kennis van scholieren

Om te begrijpen wat voor kennis de scholieren hebben verkregen kijken we naar de enquête, genaamd "Learner reports". Hierbij wordt er gevraagd naar wat de scholieren hebben geleerd in de vorm van open vragen. De enquête is gesplitst tussen vier vragen waarbij de scholieren werden gevraagd werden om een zin aan te vullen. Van de learner reports zijn de resultaten van 35 scholieren beschikbaar.

Van de 35 scholieren, hebben 16 scholieren aangegeven dat ze iets nieuws hebben geleerd over Scratch. Hierbij hebben scholieren aangegeven dat ze beter met Scratch kunnen werken of een bepaalde functie hebben leren kennen waardoor ze Scratch beter kunnen gebruiken. Eén scholier geeft als antwoord bij de vraag "Ik heb geleerd dat" : *"In scratch ook kan inspreken en de stemtjes kan aanpassen (Lager of hoger maken)"*.

Van dezelfde 35 scholieren hebben maar vier scholieren aangegeven dat ze iets nieuws hebben geleerd over de taal Engels. Hierbij meldt één scholier : *"Je moet overal in de zin de i met een hoofdletter schrijven"*. Eén van de vier scholieren behoort tot de 16 scholieren die hebben aangegeven dat ze ook iets nieuws met Scratch hebben geleerd.

Frequency (n)				
Vragen	Antwoord "Ja"	Antwoord "Onzeker"	Antwoord "Nee"	Totaal
"Ik heb genoten van de les"	30	4	0	34
"Ik vond de les interessant"	24	7	3	34
"Ik heb begrepen wat ik deze les moet doen"	26	8	0	34
"Ik heb de stof van vandaag begrepen"	25	7	0	32
"Ik vond de les moeilijk"	0	12	22	34

Table 7: Resultaten van de exit tickets van 16 november

Frequency (n)				
Vragen	Antwoord "Ja"	Antwoord "Onzeker"	Antwoord "Nee"	Totaal
"Ik heb genoten van de les"	37	2	0	39
"Ik vond de les interessant"	32	6	1	39
"Ik heb begrepen wat ik deze les moet doen"	37	2	0	39
"Ik heb de stof van vandaag begrepen"	30	8	1	39
"Ik vond de les moeilijk"	5	6	28	39

Table 8: Resultaten van de exit tickets van 23 november

4.1.5 Gedrag & moeilijkheden

Voor de tweede en derde week moesten de scholieren enquêtes invullen gebaseerd op hun ervaringen. De enquêtes, genaamd "Exit tickets", focussen zich op de houding van de studenten, de obstakels in de lessen en het begrijpen van de lesstof. Tabel 7 toont de resultaten van de tweede week, waarbij de CT-integratie lessen werden geïntroduceerd. Tabel 8 toont de resultaten van de derde week. De "Exit tickets" bevatten ook open vragen, maar veel scholieren heb dit niet ingevuld. Voor de tweede week hadden paar scholieren, die de open vragen hadden ingevuld, genoteerd hoe leuk Scratch was. Eén scholier had genoteerd dat Scratch afleidend was. Voor de derde week waren er geen open vraag antwoorden.

Voor het begrip van de scholieren letten we op de vragen "Ik heb begrepen wat ik deze les moest doen." en "Ik heb de stof van vandaag begrepen". In de tweede week, heeft geen scholier "Nee" beantwoord op deze twee vragen. Een meerderheid hebben de vragen positief beantwoord, waarbij een minderheid "Onzeker" hebben beantwoord. De derde week heeft soortgelijke verhouding qua antwoorden. Van de vier scholieren die hadden aangegeven dat ze geen programmeerervaring hadden, hebben drie daarvan de Exit tickets ingevuld. Alle drie hebben ook aangegeven dat ze de stof begrepen en wisten wat ze moesten doen in de les. Alleen één scholier heeft aangegeven dat ze de stof niet heeft begrepen.

Voor de obstakels tijdens de CT-geïntegreerde Engelse lessen wordt de focus gelegd op de vraag "Ik vond de les moeilijk". In de tweede week zien we een meerderheid die "Nee" hebben beantwoord. De overige scholieren hebben "Onzeker" beantwoord. Voor de derde week zien we een kleinere minderheid die "Onzeker" hebben beantwoord, maar ook een verhoging van scholieren die "Ja"

hebben beantwoord. Van die 5 scholieren, hebben drie aangegeven dat ze de stof hebben begrepen en aangegeven wat ze moesten doen in de les. De overige twee scholieren hebben als antwoord “Onzeker” gegeven bij de vraag “Ik heb de stof van vandaag begrepen”.

Voor de houding van de scholieren kijken we naar de antwoorden van de vragen “Ik heb genoten van de les” en “Ik vond de les interessant”. Voor de tweede week zien we positieve resultaten waarbij een grote meerderheid heeft genoten van de les. Een lagere meerderheid geeft aan dat de lessen ook interessant waren. Van de drie scholieren die hebben aangegeven in de tweede week dat ze de les niet interessant vonden, hebben wel aangegeven dat ze de lessen niet moeilijk vonden en de stof hebben begrepen. De derde week toont soortgelijke resultaten, waarvan een meerderheid positief hebben beantwoord op beide vragen. Dit keer heeft één persoon “Nee” beantwoord bij de vraag “Ik vond de les interessant”. Ook deze scholier heeft aangegeven dat die de les niet moeilijk heeft.

Bij de enquête “Learner reports” is er ook gekeken naar de houding waarop scholieren de vragen hebben beantwoord. We willen weten of de scholieren een positiever beeld hebben van de CT-geïntegreerde lessen. Hierbij letten we op negatieve eigenschappen die de scholieren vermelden wanneer ze de vragen beantwoorden. Ook kijken we naar de antwoorden die een positiever beeld van de CT-geïntegreerde lessen signaleren. Bij deze sectie wordt de focus gelegd naar de positieve of negatieve beeld die de scholieren beschrijven in hun antwoorden. Er wordt gelet wat de scholieren makkelijker en leuker vinden vergeleken met wat ze hebben geleerd.

Bij de positieve antwoorden, wordt er gelet naar het zelfvertrouwen die de scholieren signaleren over wat ze hebben geleerd. Van de 35 scholieren, hebben 17 scholieren een positiever beeld gekregen over hetgeen wat ze hebben geleerd. Hierbij signaleren de scholieren dat ze een bepaalde onderwerp die werd meegegeven in de CT-geïntegreerde lessen makkelijker of veel leuker vonden. Sommige scholieren signaleerde hoe ze meer zelfvertrouwen in een bepaalde onderwerp hadden. Tabel 9 toont hoe vaak een bepaalde commentaar werd vermeld van de 18 scholieren.

Vermeldingen	Frequency (n)
Programmeren makkelijker	13
Programmeren leuker	3
Engels makkelijker	1
Andere schoolvakken leuker	1

Table 9: Hoe vaak er werd gesignaleerd naar hoe makkelijk of leuk een bepaalde onderdeel is

Eén scholier vermeldt: *“Eerst dacht ik dat ik programmeren stom vond, maar eigenlijk vind ik het best leuk. Misschien ga ik het ook in m’n vrije tijd doen”*. Een ander scholier heeft het volgende geschreven : *“Ik heb geleerd dat het niet waar is dat ik slecht met computers ben”* en ook : *“Ik heb geleerd dat ik goed kan programmeren”*.

Er waren ook scholieren die negatieve antwoorden gaven. Hierbij gaven de scholieren aan dat ze een bepaalde onderwerp moeilijk of niet boeiend vonden. Van de 35 scholieren gaven 10 scholieren een negatieve beeld over hun leerervaring. Tabel 10 toont hoe vaak een scholier hun leerervaring hebben beschreven door middel van antwoorden in een negatieve manier. Hierbij kunnen meerdere punten,

zoals beschreven in de tabel, genoemd worden door een scholier. Eén scholier vermeldt : *“Ik heb geleerd dat ik slecht ben in programmeren”*. Een ander scholier heeft vermeldt : *“Ik heb geleerd dat ik slecht ben in scratch”*, maar hierbij schreef hij ook wat hij allemaal geleerd heeft waaronder : *“Programeren op scratch, Engelse gramatica, Engelse woorden, Engelse zinnen”*.

Vermeldingen	Frequency (n)
Programmeren moeilijk	10
Programmeren niet leuk	1
Programmeren duurt te lang	1
Helemaal niet kunnen programmeren	1

Table 10: Hoe vaak er werd gesignaleerd naar hoe moeilijk of niet leuk een bepaalde aspect is

Van de 10 scholieren hadden 5 scholieren daadwerkelijk opgeschreven dat ze iets hadden geleerd uit de lessen.

4.2 Computational thinking vaardigheden

In deze sectie kijken we naar de specifieke "computational thinking" vaardigheden die zijn toegepast in de lessen. We kijken ook wat voor nieuwe vaardigheden zijn verkregen bij het project. Om dit te achterhalen is er gebruik gemaakt van de scholieren interviews en de enquêtes "Learner reports" en "Praktische opdracht".

4.2.1 Algoritmisch denken

In deze sectie kijken we naar het "Algoritmisch denken" vaardigheid. Hierbij kijken we of de scholieren hun project in stappen kunnen verwerken en zo hun eigen "Algoritme" bouwen om een duidelijk proces te vormen.

Om de vaardigheden te analyseren, zijn er specifieke vragen gesteld over het algoritmisch denken van de scholieren. Bij de enquête "Praktische opdracht" moesten de scholieren een algoritme bedenken die hun digitale scratch animatie beschreef. De scholieren moesten de vraag beantwoorden door middel van een als-dan redentatie. De als-dan redentatie refereert een "if-then" statement wat vaker voorkomt als een methode voor programmeren. De vaardigheid Algoritmisch denken is vereist voor een goed structuur voor dergelijke redentaties. Ondanks de "als-dan" redentatie, kijken we ook naar de stappen die de scholieren moeten maken. Hun process splitsen in stappen is ook een aspect van algoritmisch denken. Van de "Praktische opdracht" zijn er 21 resultaten beschikbaar.

Voor deze vraag zijn er diverse antwoorden. Hierbij kijken we naar antwoorden die het project in logische stappen uitleggen en/of gebruik maken van een logische als-dan redentatie. Van de 21 resultaten zien we dat 10 scholieren hieraan voldoen. Van de overige scholieren zijn de antwoorden kort, onduidelijk of geen juiste antwoord op de vraag.

Eén scholier heeft een als-dan redentatie gebruikt bij zijn/haar uitleg : *"Kat loopt. Komt tegen een goblin. Ze introduceren zichzelf. Kat praat eerst. Als hij klaar is, dan begint goblin. Zo gaat het door."*. Het antwoord is in stappen uitlegd en er is gebruik gemaakt van een als-dan redentatie. Sommige scholieren hebben de vraag beantwoord in stappen, maar zonder als-dan redentatie, zoals één scholier heeft vermeldt : *"Meisje valt van de steen. Andere meisje helpt haar. Ze leren elkaar kennen. Een meisje die ze kent komt ook opeens. Ze gaan iets leuks doen."*. In dit geval, wordt haar proces in stappen verdeeld.

Voor de scholieren interviews werd ook de vraag gesteld wat voor stappen de scholieren nemen voor het project. Hiervan zijn er 6 antwoorden beschikbaar. Van de 6 scholieren hebben 5 vermeldt dat ze eerst op papier alles uitschrijven en daarna begonnen met het programmeren. Eén scholier had hierbij vermeldt, op de vraag "Welke stappen volg jij om je project te maken" : *"Als eerst ging ik alles inschrijven. Als eerst ging ik eigen script maken. Wat ik allemaal moest zeggen en schrijven. En daarna ging ik alles gewoon schrijven en dan ging ik thuis alles inspreken"*. De overige scholier had vermeld dat ze gelijk begon met het programmeren van het project. De scholier had hierbij ook vermeld dat ze naar oude projecten heeft gekeken als ondersteuning.

4.2.2 Generalisatie

Voor deze sectie kijken we naar de vaardigheid "Generalisatie". Hierbij kijken we of de scholieren delen van voorafgaande oplossingen kunnen hergebruiken en toepassen in de obstakels van het experiment. Om te zien of scholieren daadwerkelijk delen van een oplossing kunnen hergebruiken voor andere probleem, oftewel in andere projecten, heeft de onderzoeker specifieke vragen gesteld aan de scholieren over deze vaardigheid. De onderzoeker vroeg tijdens de interviews of de scholieren hun project in andere project kunnen hergebruiken.

Bij 3 interviews hebben we duidelijk antwoord gekregen op de vraag. Eén scholier was niet zeker of delen van het project hergebruikt kunnen worden. De andere twee scholieren melden hoe ze daadwerkelijk delen kunnen hergebruiken als het om dezelfde onderwerp gaat, anders niet. Van die twee scholieren, meldt één dat het hergebruiken moeilijk kan zijn in bepaalde situaties. Deze scholier verwijst naar een project van een ander scholier en meldt : *"Of dan zou je zo'n andere kind in de klas heeft zo'n lange code geschreven. Heb je die van (Scholier) gezien? Die heeft hele lange code wat soms iets moeilijk is want je weet niet waar je bent.[...] want als het heel lang is dan weet je niet welke deel het is. Dat zal waarschijnlijk moeilijker zijn."* De scholier toont hoe hergebruik bij bepaalde projecten lastiger kan zijn dan anderen. Andere scholieren melden niet uit zichzelf of ze delen van hun project zullen gebruiken voor toekomstige projecten

4.2.3 Evaluatie

Voor het experiment bekijken we hoe de scholieren hun eigen project evalueren. We verdiepen ons hoe ze hun project evalueren maar ook hoe ze terugkijken naar hun aanpak en oplossingen. Hierbij kijken we naar de resultaten van de enquête "Praktische opdracht" en ook naar de interviews waarbij de scholieren vragen werden gesteld over hun project.

Bij de enquête "Praktische opdracht" werd de vraag gesteld of de scholieren tevreden zijn met het project in de vorm van een open vraag. Ze konden ook vermelden waarom wel of waarom ze niet tevreden zijn. Hiervan zijn er 22 resultaten beschikbaar. Hierbij gaven 19 resultaten aan dat ze tevreden waren met hun project. Van de 3 overige resultaten melden 2 daarvan dat ze onzeker waren en 1 gaf een "Nee" als antwoord. Er werden een paar verbeteringspunten genoemd bij de 19 positieve antwoorden. De verbeteringspunten gingen niet om grote problemen, maar meer om kleine ongemakken. Zo heeft één scholier vermeld : *"Ik ben er wel tevreden mee alleen ik heb niet zo'n Engels accent dus het klinkt een beetje raar."*

Bij de interviews had de onderzoeker een soortgelijke vraag gesteld. Hiervan zijn er 5 resultaten beschikbaar. Uit alle resultaten kwamen de scholieren positief over. 2 scholieren waren absoluut tevreden met hun project en wouden niks veranderen. 1 scholier was nog niet klaar met zijn project en vermeldde dat de scholier zich vooral ging focussen op fouten na afloop. De 2 overige resultaten hadden een identiek opmerking. Beide hebben vermeld hoe de grafische aspect van de animatie een probleem voor ze vormde. Hierbij vermeldde de scholier : *"De personages zijn echt slecht. Dus ik denk dat ik gewoon een foto ga maken of zoiets als staand, en dan naar e-mail sturen en dat personage maken en ja, het moet bewegen[...]Ik denk dat het goed is, maar de personages zijn slecht"* ["The characters are really bad. So I think I'm just gonna make a picture or something like standing and then send to mail and make that character and yeah it needs to move[...]I think its good

but the characters are bad”]. De ander scholier die ook vermeldt had dat ze iets grafisch wouden veranderen had een soortgelijk antwoord. Hierbij beantwoordde de scholier de vraag wat de scholier wou veranderen. De scholier gaf dit als antwoord : *“De karakters want ik was aan het tekenen maar het gaat moeilijk om het in scratch te zetten want het gaat er anders uitzien. En ook al teken je in scratch, zou het anders eruit zien in de achtergrond.”*.

4.2.4 Decompositie

Bij decompositie kijken we naar de manier waarop scholieren problemen oplossen. Door middel van een probleem te splitsen in kleinere problemen pas je een computationeel denkende vaardigheid toe. Hierbij kijken we naar de enquête “Praktische opdracht” en de scholieren interviews.

Bij de enquête “Praktische opdracht” werd de vraag gesteld of de scholieren vast kwamen te zitten tijdens het programmeren, en als het zo was, wat hadden ze gedaan om het op te lossen. Hiervan zijn er 24 resultaten beschikbaar. Hierbij kijken we vooral naar de oplossingen of de scholieren daadwerkelijk het probleem opsplitsen en zo kunnen oplossen. Van de 24 resultaten hebben 10 scholieren “Nee” beantwoord op de vraag. Van de overige 14 resultaten hebben 13 scholieren “Ja” vermeld en de resterende scholier vermeldde dat hij nog niet klaar was. Van de 13 scholieren die “Ja” hadden vermeld waren er verschillende problemen die tevoorschijn kwamen, zoals een opname probleem of gebrek aan kennis. Eén specifieke oplossing kwam wel vaker terug bij de meeste scholieren. Van de 13 scholieren die “Ja” hadden vermeld waren er 7 scholieren die er ook bij vermeldde dat ze hulp ging zoeken voor hun probleem, zoals één scholier vermeldt : *“We konden iets niet opnemen, dus hebben we iemand geroepen”*. Van de overige 6 scholieren heeft maar één scholier een eigen oplossing vermeld, waarbij de scholier vermeld bij zijn antwoord : *“Soms ging het praatwolk ergens anders naartoe maar dan moest je gaan refreshen”*. Van de 24 beschikbare resultaten heeft niemand vermeld hoe ze het probleem hebben vereenvoudigt.

Bij de interviews zien we de toepassing van decompositie ook niet terug. Hierbij had de onderzoeker dezelfde vraag gesteld aan de scholieren. We hebben gekeken naar zeven verschillende antwoorden. vijf van de zeven scholieren vermeldde dat ze om hulp hadden gevraagd of door middel van ”tutorials” oplossingen konden vinden. De overige twee scholieren hadden een soortgelijke probleem waarbij een bepaalde gedeelte verkeerd of verwijderd was. In beide gevallen hebben ze het problematische gedeelte opnieuw geschreven. Geen scholier van de interviews heeft vermeld dat degene decompositie toepassen.

4.3 Pedagogical Content Knowledge

Om de pedagogische kennis van de docent te achterhalen, werd de docent aan het eind van het onderzoek geïnterviewd door één van de onderzoekers. De eerste vier subsecties bespreken, respectievelijk, de PCK-componenten die werden vermeld bij de interview. De PCK-componenten zijn vooral gericht op de leerdoelen (M1), het begrip van de scholieren (M2), De instructionele strategieën die zijn toegepast (M3) en de evaluatie van de lessen (M4).

4.3.1 M1 : Leerdoelen

Voor de component M1, kijken we naar de kennis over doelen en doelstellingen van het onderwijzen van de CT-geïntegreerde lessen. We willen achterhalen wat de docent wilt onderwijzen aan de scholieren en wat voor kennis de scholieren moeten opdoen dankzij de CT-geïntegreerde lessen.

Bij het interview werd er gevraagd over de leerdoelen van de scholieren. Hierbij werd de focus gelegd op de computationele denkende vaardigheden en Engelse aspecten. Wanneer de docent gevraagd werd over de globale leerdoelen van het experiment, vermeldde de docent : *"Kortom, dat ze het vocabulaire en de grammatica integreren in het creëren van een animatieverhaal en dat op basis van het programma scratch. Dus gebruik de stappen die je nodig hebt in het begin, dus maak in feite het algoritme helemaal opnieuw om een verhaal te maken"* [*"Basically, that they integrate the vocabulary and the grammar into creating an animation story and that based on the program scratch. So using the steps u need in scratch so basically creating the algorithm in scratch to create a story"*]. Hierbij wil de docent dat de scholieren een animatie maken met de programmeertaal Scratch door middel van een algoritme toepassen en hun woordenschat en grammatica. De docent werd ook gevraagd over leerdoelen die te maken hebben met Engelse aspecten. De docent had hierbij vermeld : *"[...] ze waren in staat om te handelen en vragen te beantwoorden in de present simple. Ze praten over hun hobby's en stellen vragen over hobby's en praten over datum en tijd, praten over plaatsen, beschrijven plaatsen en gebruiken de woordenschat die ze hebben geleerd met de grammatica of de verhalen"* [*"[...] they were able to act and answer questions in the present simple. They talk about their hobbies and ask questions about hobbies and talk about date and time, talk about places, describe places and use the the vocabulary that they learned with the grammar or the stories"*]. De scholieren moesten de onvoltooide tegenwoordige tijd toepassen in hun animatie en praten over algemene onderwerpen zoals hobby's en plaatsen. Bij de laatste antwoord, vermeldt de docent dat zijn leerdoelen de woordenschat en de grammaticale toepassing van de onvoltooid tegenwoordige tijd zijn.

4.3.2 M2 : Begrip van scholieren

Bij de component M2, kijken we hoe de scholieren de CT-geïntegreerde lessen hebben ervaren vanuit het perspectief van de docent.

De interviewer had zich verdiept naar de reacties van de studenten. Wanneer er werd gevraagd hoe de scholieren de lessen hadden ervaren, had de docent dit vermeld : *"Veel van hen, ze vonden het erg leuk.[..] Ze dachten dat het moeilijk was, maar ze zeiden dat ze het leuk vonden. [...] want er is veel vaardigheid wat ik heb opgemerkt. Als je dat doet, is er veel vaardigheid vereist [...] Je moet ook plannen om samen te kunnen werken en ook dan de hele kras hoe programmeer je die dingen, dus het zijn veel vaardigheden die samen moeten komen om iets te creëren, ik Raad eens. Dat was het moeilijkste voor hen[...]"* [*"Alot of them, they liked it alot.[..] They thought it was hard but they said they liked it. [...] cause there is alot of skill what i noticed. When doing it is that a lot of skill is required [...] You also need to plan able to work together and also then the whole scratch how do you program those things so its alot of skills that have to come together to create something, I guess. That was the hard part for them[...]"*]. De docent vermeld hierbij dat de scholieren het grotendeels leuk vonden maar de lessen waren wat moeilijker dan normaal voor de scholieren. Veel vaardigheden zijn vereist voor het project, niet alleen Engels kunnen maar ook teamwerk en computationele

denkende vaardigheden. Programmeren was ook een obstakel voor bepaalde scholieren zoals de docent heeft vermeld.

4.3.3 M3 : Instructionele strategieën

Voor de component M3, kijken we wat voor strategieën worden toegepast om les te geven. We kijken naar hoe de integratie van CT heeft plaatsgevonden vanuit zijn perspectief en wat voor verschillen er zijn tussen de CT-geïntegreerde lessen en de standaard manier van lesgeven.

De interviewer heeft gevraagd naar instructionele strategieën die werden toegepast tijdens het onderzoek. Eerst werd er gevraagd hoe de docent, voor het project, les gaf aan de scholieren. De docent had hierbij vermeld : *"[...] dus ik gebruik de directe instructies-methode, dus expliciete woordenschat aanleren en wat we doen is dat we ze grammatica leren, dan doen we in feite een test"* [*"[...] so I use direct instructions method, so explicit teaching vocabulary and what we do is we teach them grammar then we do a test basically"*]. Dit is een standaard manier van lesgeven wat voorkomt in meeste middelbare scholen. "direct instructions methods" is de standaard methode voor docenten, waarbij ze voor het klaslokaal staan en de lesstof presenteren. Hierbij geeft de docent expliciete, begeleide instructies aan de scholieren.

Daarna werd er gevraagd hoe de docent daadwerkelijk "Computational thinking" integreert in zijn lessen, door middel van instructionele strategieën. Hierop had de docent geantwoord : *"[...] wat we normaal gesproken zouden laten doen is, je weet wel, geef ze een schrijfo opdracht.. dit keer deden we met een digitaal verhaal. Je zou het verhaal moeten plannen en maken op basis van het maken van een algoritme om dat verhaal in principe compleet te maken "* [*"[...] what we normally would let's say do is, you know give them a writing assignment.. this time we did with a digital story. You would need to plan and create the story based on creating an algorithm for that story to be complete basically"*]. Hierbij vermeldt hij dat er gebruik werd gemaakt van een digitale verhaal in plaats van een schrijf opdracht. De interviewer voegde daarbij toe hoe een andere toegepaste strategie, een storyboard in een verhaal veranderen, kon zijn. Hierbij was de docent het ermee eens. De docent had geen bepaalde "Computational thinking" vaardigheden vermeld bij zijn antwoord.

De docent werd gevraagd hoe de docent kon bepalen of de scholieren hun leerdoelen hebben bereikt. De docent had bij het antwoord dit vermeld : *"[...] dus moesten ze voice-overs doen en als het op creativiteit aankomt, ga ik kijken, zoals hoe heb je je verhalen eigenlijk gemaakt, zijn er bewegende elementen? Hoe ging het praten? [...] dat is waar ik het algoritme-gedeelte ga controleren, zoals hoe heb je dat gedaan, hoe ziet het eruit?"* [*"[...] so they had to do voice overs and when it comes to creativity thats when im gonna check like how did you actually make your stories, is there any moving elements? How did the talking go? [...] thats where im gonna check the algorithm part like how did you do that, how does it look?"*]. De docent vermeld hierbij niet veel over de Engelse leerdoelen, maar focust vooral op de computatonele denkende onderdelen. Hij kijkt niet alleen naar de eindresultaat maar ook naar het proces van de scholieren.

De docent werd ook gevraagd wat de verschillen waren tussen de standaard manier van lesgeven en de "Computational thinking" integrerende manier van lesgeven. Hierbij had de docent vermeld :

”Het grootste verschil is dat er meer vaardigheid vereist is, maar de standaard is geïsoleerd van de grammatica en de woordenschat, maar hiermee wordt de hele planning en samenwerking één, dus dat maakt het uitdagender voor de studenten” [“*The biggest difference is that more skill is required, but the standard one is isolated to the grammar and the vocabulary but with this the whole planning and working together becomes one so that makes it more challenging for the students*”]. De docent vermeldt hoe scholieren meer vaardigheden moeten toepassen in hun manier van lesgeven. Dit zijn niet alleen computationele denkende vaardigheden, maar ook een vorm van teamwerk wordt hierbij toegepast.

4.3.4 M4 : Evaluatie

Voor de component M4, kijken we naar de evaluatie van de docent. We willen een beeld schetsen hoe hij de lessen heeft ervaren, wat voor moeilijkheden hij tegenkwam en hoe effectief de docent de CT-geïntegreerde lessen zelf vindt.

De docent werd allerlei vragen gesteld over zijn eigen perspectief van de lessen. De eerste vraag die de interviewer had gesteld ging over de obstakels die de docent tegenkwam tijdens de computationeel denkende geïntegreerde lessen. De docent noemde twee problemen op. Voor de eerste probleem had de docent vermeld dat er niet genoeg computers waren. Dit is meer een technische probleem wat kan verschillen per school. Hiervoor zijn er natuurlijk wat oplossingen voor, maar er werden geen oplossingen voor dit probleem vermeld door de docent. Bij het tweede probleem gaf de docent meer commentaar : *”[...] als leraar moet je meer kennis hebben van welk programma je ook gaat gebruiken om de studenten beter te begeleiden en deze keer heb ik dat gecompenseerd door alle studenten te gebruiken, maar het is iets wat je hebt. Dat was praktisch. Het werkte goed omdat ik die studenten heb, maar ik denk dat je als docent meer kennis moet hebben over de programmering om de studenten beter te begeleiden.*” [“*[...] as a teacher you need to be more knowledgeable whatever program you are gonna use in order to guide the students better and this time, I compensated that by using all the students but its something that you have. That was a practicality. It worked out well because i have those students, but I think as a teacher you should be more knowledgeable about the programming to guide the students better.*”]. Hij vermeldt hoe de kennis van een docent een obstakel kan vormen op daadwerkelijk leerzame opdrachten aan te bieden voor de studenten. Los van de vraag gaf de docent ook extra commentaar erbij. De docent gaf als commentaar : *”[...] Ik denk dat om het goed te doen, ik mijn niveau van computationeel denken en programmeren in het algemeen moet verhogen om [...] goede opdrachten voor hen te kunnen ontwerpen.*” [“*[...] I do think that in order to do it well, I need to up my level of computational thinking and programming in general to be able to make [...] to design good assignments for them.*”]. De docent benadrukt de achtergrond kennis die nodig was om daadwerkelijk zulke vaardigheden aan te leren aan de scholieren.

De docent werd ook gevraagd hoe de scholieren de Engels talige leerdoelen en de Computationele denkende leerdoelen hebben geleerd vergeleken met de standaard manier van lesgeven. Hierbij gaf de docent een gemixt reactie : *”Ik moet ze nog corrigeren, ik ben benieuwd, ik zag een aantal goede dingen maar ik zag ook een aantal waar ik aan dacht, ahh ik weet niet of dit gaat lukken. Ik ben nieuwsgierig. Ik kan die vraag nog niet helemaal beantwoorden. Ik heb niet overal zicht op alle klassen die het al hebben gedaan, ik moet ze nog corrigeren.*” [“*I still have to correct them, I’m curious to see, i saw some good things but I also saw some where i was thinking, ahh i dont know*”].

if this will work out. I'm curious. I cant really answer that question completely yet. I dont have all around view of all the classes that have done it yet, I still gotta correct them.”]. De docent ziet positieve maar ook negatieve aspecten in het nieuwe manier van lesgeven. De docent had nog iets positiefs vermeld over het computationele denkende gedeelte, namelijk : *”de positieve noot is dat we coderen op onze school en de codeerleraar liep toevallig rond en hij was erg enthousiast om ze zo te zien werken. Hij zei voor volgend jaar dat hij wil samenwerken en iets voor de ander wil doen, dus dat zou interessant moeten zijn ”*[*“the positive note is that, we have coding in our school and the coding teacher happened to walk around and he was very enthusiastic seeing them work like that. He said for next year he wants to collaborate and do something to the other so that should be interesting”*]. De combinatie van Computationele denkende geïntegreerde lessen en een codasium school past goed bij deze specifieke manier van lesgeven.

4.4 Implementatie suggesties van docent

Wanneer de docent werd gevraagd wat voor verbeteringen toegepast kunnen worden, had de docent meerdere kleine onderdelen genoemd wat verbeterd kon worden, zoals de templates die werden gebruikt. Eén verbetering wat de docent benadrukte was vooral de planning, hierbij vermeldt de docent : *”Veel studenten wilden programmeren voordat ze een script maakten, weet je, een script maken is niet leuk, maar het programma is veel leuker, dus in die zin zou de hele planning beter kunnen zijn [...] door ze de tools te geven die ze kunnen eigenlijk alles beter plannen en ze meer gedetailleerde stappen geven. Creëer meer stappen voor hen [...] Ik denk dat de sprong van Engels naar computationeel denken niet al te groot was, maar het had veel soepeler kunnen zijn dan het was, ik denk dat dat ook iets is dat ik leuk vind”* [*“A lot of student wanted to program before making a script, you know, making a script is not fun but the program is way more fun, so in that sense the whole planning could be better [...] giving them the tools that they can actually plan everything better and give them more detailed steps. Create more steps for them [...] I think the jump from english to computational thinking was not too big but it could have be way more smoother then it was, I think that would be something that I like too”*]. Voor de docent konden de stappen bedoeld voor de scholieren verbeterd worden. Volgens de docent, waren bepaalde scholieren meteen begonnen met programmeren waardoor hun structuur voor het hele proces wat onduidelijker werd. Hierdoor kunnen de scholieren zelf obstakels ervaren.

Aan het eind van het interview werd er nog gevraagd of de docent de integratie van Computationeel denken in de toekomst terug zien komen in zijn lessen. De docent gaf een interessant antwoord op de vraag : *”Ja, ik denk gewoon in welk jaar het het beste zou zijn om het praktisch te doen, want de studenten vonden dit erg leuk, maar het was veel organiseren, maar ik weet niet of alles aanwezig is. Als de infrastructuur er is, is er natuurlijk minder organiseren. [...] het kost veel energie laat ik het zo zeggen”*[*“Yeah, I'm just thinking in what year it would be the best to do it practicality, cause the students really enjoyed this but it was a lot of organizing but I dont know if everything is there. If the infrastructure is there then there is less organizing of course. [...] it costs alot of energy let me put it like that”*]. De docent was wel positief over de ervaring i.v.m. met de scholieren, maar het organiseren ervan was een uitdaging. De docent vermeldt dat het proces in de toekomst waarschijnlijk wat beter gaat.

5 Conclusie

Om de hoofdvraag te beantwoorden, beantwoorden we eerst de deelvragen om een goed beeld te vormen over de effectiviteit van de CT-geïntegreerde lessen.

Bij de eerste deelvraag "Wat zijn de houdingen van studenten ten opzichte van "Computational thinking" geïntegreerde Engelse lessen?" hebben we een positief antwoord gevonden. Grotendeels van de scholieren hebben genoten van de CT-geïntegreerde lessen. Uit de resultaten is te zien dat er daadwerkelijk interesse is voor de CT-geïntegreerde lessen. Veel scholieren hebben ook gemeld dat ze de lessen niet moeilijk vonden. Dit is wel logisch, omdat meeste scholieren ook ervaringen hebben gehad met scratch of andere programmeertalen. Bij opmerkingen of het programmeren moeilijk of makkelijk is, waren de aantallen ongeveer gelijk. Veel van de scholieren hebben ook de lessen kunnen begrijpen. In het algemeen hebben de scholieren gemeld dat ze de CT-geïntegreerde lessen leuk en leerzaam vonden. We zagen nauwelijks negatieve opmerkingen over de lessen. Van de weinig negatieve opmerkingen ging het vooral om de moeilijkheid van programmeren, wat wel te begrijpen is. Bij de houding tegenover de CT-geïntegreerde lessen kunnen we een positief beeld concluderen.

Voor de deelvraag "Hoe dragen de "Computational thinking" geïntegreerde lessen bij aan de "Computational thinking" gerelateerde kennis van studenten" zien we bepaalde "computational thinking" vaardigheden terug komen. Bij de computationeel denkend vaardigheid "Algoritmisch denken" zien we wel aspecten ervan terug. De scholieren gebruiken daadwerkelijk stappen voor hun digitale animatie. De antwoorden verschillen in de grootte van de stap. Sommige scholieren beschrijven hun proces in weinig maar grote stappen terwijl andere scholieren uitgebreider zijn met kleinere maar meer stappen. We zien dat algoritmisch denken wel toegepast wordt in de CT-geïntegreerde lessen.

Voor generalisatie keken de scholieren kritisch naar het hergebruiken van oplossingen. Grotendeels vermeldde de scholieren dat het vooral werd gedaan zolang het om dezelfde onderwerp ging. In bepaalde situaties, zoals ingewikkeld en lang code, werd dit niet aangeraden.

We zien wel evaluatie terug bij de CT-geïntegreerde lessen. Veel van de scholieren waren tevreden met hun project. We zagen bepaalde opmerkingen die vooral gebaseerd waren op het Engels- en programmeer onderdelen. Van de minderheid die niet tevreden waren, werden er vooral opmerkingen gegeven over het Engelstalig gesprek in de animatie of de grafische kant van het project.

Bij de computationeel denkende vaardigheid decompisitie zien we de implementatie ervan niet terug. Geen scholier heeft vermeld dat ze een bepaalde probleem opsplitsen en zo oplossen. We zagen ook geen duidelijk implementatie van decompisitie terug toen het direct gevraagd werd. Grotendeels van de scholieren hebben simpelweg hulp gezocht wanneer ze een probleem hadden.

Voor de deelvraag "Hoe dragen de "Computational thinking" geïntegreerde lessen bij aan de Engelse kennis van studenten" zien niet veel van terug in de resultaten. Weinig scholieren hadden gemeld dat ze wat nieuws hebben geleerd over de taal Engels. De resultaten toonde hoe meer scholieren hadden geleerd over "Computational thinking" en programmeren dan de taal Engels.

Voor de deelvraag "Wat is de Pedagogische Content Knowledge (PCK) van de docent met betrekking tot "Computational thinking" integratie?" zien we veel interessante punten van de docent. Bij de leerdoelen van de docent zagen we dat zijn focus vooral lag op de woordenschat en grammaticale

kennis. Hij vermeldt ook hoe hij de manier van algoritmes toepassen wilt aan leren aan de scholieren. De docent was ook positief over zijn perspectief van de houdingen en begrip van de scholieren. Hij vermeldde dat veel scholieren het leuk en interessant vonden, maar dat het wel moeilijk was voor bepaalde scholieren. Hij was neutraal over de integratie van Engels en computationeel denkende leerdoelen waarbij hij positieve en negatieve opmerkingen erover had. Meeste negatieve opmerkingen lag aan de moeilijkheid van de integratie. Hij vond wel de integratie beter gaan omdat het heeft plaatsgevonden in een school waar coderen wordt lesgegeven.

Over de strategieën van lesgeven waren er interessante opmerkingen die de docent had vermeld. Hij vond dat de nieuwe manier van lesgeven de scholieren veel meer kon aanleren, niet alleen Engels maar ook computationeel denkende vaardigheden. Hij had hierbij vermeld dat het wel lastiger is om les te geven in de nieuwe manier vergeleken met de oude manier. Hij benadrukt hoe er meer kennis van de docent is vereist om op de juiste manier les te geven.

Bij de evaluatie kwamen veel constructieve opmerkingen voor. De docent vermeldde vooral hoe een docent meer ervaring en/of kennis nodig heeft om daadwerkelijk CT-geïntegreerde lessen te implementeren. Hij vermeldde hoe betere planning en structuur was vereist om goed les te geven aan de scholieren. De laatste vraag vermeldde een interessant punt. De docent concludeert hoe hij de CT-geïntegreerde lessen nuttig vond maar onzeker was over de tijd waarop CT-geïntegreerde lessen daadwerkelijk geïmplementeerd moesten worden. Hij vermeldt dat in de toekomst het proces veel beter gaat.

Met behulp van de deelvragen kunnen we de hoofdvraag "Hoe effectief is de integratie van computational thinking in het schoolvak Engels door middel van digital storytelling in het Nederlands onderwijs?" beantwoorden. We kunnen hieruit concluderen dat de scholieren bepaalde computationeel denkende vaardigheden implementeren, zoals evaluatie, algoritmisch denken en generalisatie. De scholieren en docent vertonen een positieve interesse in de CT-geïntegreerde lessen. Door middel van "Digital storytelling" vinden de scholieren veel plezier in het man van hun project. Maar de computationele vaardigheid "Decompositie" werd niet uit zichzelf geïmplementeerd in de lessen. Ook is de moeilijkheidsgraad van de lessen een obstakel voor bepaalde scholieren en soms ook voor de docent. De resultaten tonen dat de CT-geïntegreerde lessen leerzaam en productief waren, maar "Computational thinking" kennis van de docent is vereist om effectief les te geven.

6 Discussie

6.1 Reflectie op resultaten

Een soortgelijke artikel van Sharon Jacob [JNTG⁺18] toont positieve en negatieve resultaten over de integratie van computational thinking in het schoolvak Engels. De integratie biedt een unieke identiteit aan het lesgeven van het schoolvak "Engels" door meerdere vaardigheden toe te passen die normaal gesproken niet worden beoefend. Toch zit de moeilijkheid in de voorbereiding en kennis van de docent. Het artikel beschrijft hoe een goede kennis over computational thinking vereist is om goed les te geven aan de scholieren.

Ketelhut et. al [KMH19] verdiept hier verder in. Hierbij werd er onderzoek gedaan naar de computational thinking kennis van meerdere docent. Het onderzoek zag hoe docenten vaak de integratie van computational thinking overschatten. Ook vonden de docent dat ze niet genoeg kennis hadden om de problemen die de scholieren tegenkwamen op te lossen. Voor ons onderzoek waren er soortgelijke problemen maar omdat het in een codasium plaatsvond was er genoeg hulp voor de docent om zulke problemen op te lossen.

De Groot [Gro] vertelt over een onderzoek waarbij er werd gekeken naar de uitdagingen van integratie van computational thinking. Hierbij werd er gemeld hoe een gebrek aan technologische middelen een probleem vormde. Ook het ontbreken van computational thinking in het curriculum vormde een obstakel voor de docenten. Maar de grootste uitdaging volgens De Groot is het daadwerkelijk beoordelen of de scholieren computational thinking vaardigheden hebben geleerd en kunnen toepassen. Meeste docent focussen zich vooral op het eindproduct en niet op het proces waardoor beoordeling lastig kan zijn.

6.2 Limitaties

Bepaalde limitaties kwamen voor in het experiment. Bepaalde scholieren hadden de enquêtes onvoldoende ingevuld waardoor hun resultaten niet geanalyseerd konden worden. Andere scholieren hadden ongewone of tegenstrijdige antwoorden gegeven waardoor hun data niet gebruikt konden worden. De analyse van de data is uitgevoerd door één persoon, hierdoor kunnen er fouten zijn ontstaan tijdens het analyseren. De grootte van de groep waarvan de resultaten bekend zijn is niet groot genoeg om representatief te zijn voor de hele school. Ten slotte is het experiment uitgevoerd in een codasium school waarbij er veel programmeer hulpmiddelen beschikbaar zijn zoals genoeg computers en docenten met codeer ervaring. Dit is niet het geval bij meeste middelbare scholen.

6.3 Toekomstige werk

Voor toekomstige werk kan het experiment uitgevoerd worden in meerdere scholen met een grotere groep. Hierbij willen we een aangepaste versie van de CT-geïntegreerde lessen implementeren. Een grote probleem voor de docent was de planning en kennis van de lessen. Voor de toekomst kunnen er aanpassingen worden gemaakt met de lessen. De aanpassing horen de lessen wat eenvoudiger te maken voor de docent en de scholieren. Eén andere verschil zou de school zijn waar de lessen plaatsvinden. Het experiment is uitgevoerd in een codasium waarbij veel oudere jaars scholieren en docent codeer ervaringen hebben. Het onderzoek is betere representatief als dit wordt gedaan in een middelbare school waar coderen geen focus heeft.

References

- [Arm12] M. Armoni. Teaching cs in kindergarten: How early can the pipeline begin? *ACM Inroads*, pages 18–19, 2012.
- [AVF⁺16] C. Angeli, J. Voogt, A. Fluck, M. Webb, M. Cox, J. Malyn-Smith, and J. Zagami. Implications for teacher knowledge. *Journal of Educational Technology and Society*, pages 47–57, 2016.
- [BH19] E. Barendsen and I. Henze. Relating teacher pck and teacher practice using classroom observation. *Research in Science Education*, pages 1141–1175, 2019.
- [BHC11] D. C. Barr, J. Harrison, and L. Conery. Computational thinking: A digital age skill for everyone. *Learning and leading with technology*, 38:20–23, 2011.
- [BLVD08] A. Berry, J. Loughran, and J. H. Van Driel. Revisiting the roots of pedagogical content knowledge. *International Journal of Science education*, pages 1271–1279, 2008.
- [BMV⁺18] T. Barcelos, R. Munoz, R. Villarroel, E. Merino, and I. Silveira. Mathematics learning through computational thinking activities: A systematic literature review. *JOURNAL OF UNIVERSAL COMPUTER SCIENCE*, 24:815, 2018.
- [Bun07] A. Bundy. *Computational Thinking is Pervasive*. 2007.
- [CLW10] J. Cuny, Snyder L., and J. M. Wing. *Demystifying Computational Thinking for Non-Computer Scientists*. 2010.
- [Cod] Codasium. <http://www.codasium.nl/>.
- [Gib12] J. P. Gibson. Teaching graph algorithms to children of all ages. *Proceedings of the 17th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 34–39, 2012.
- [Gro] J. Groot. Teaching computational thinking - what do our educators need? Master-scriptie, Delft University of Technology.
- [Gro90] P. L. Grossman. *The Making of a Teacher: Teacher Knowledge and Teacher Education*. 1990.
- [JNTG⁺18] Sharin Jacob, Ha Nguyen, Colby Tofel-Grehl, Debra Richardson, and Mark Warschauer. Teaching computational thinking to english learners. 07 2018.
- [KMH19] D.J. Ketelhut, K. Mills, and E. et al. Hestness. Teacher change following a professional development experience in integrating computational thinking into elementary science. *Sci Educ Technol*, pages 174—188, 2019.
- [KNA12] KNAW. *Digitale geletterdheid in het voortgezet onderwijs: Vaardigheden en attitudes voor de 21ste eeuw*. 2012.

- [LM20] J. A. Lyon and A. J. Magana. *Computational thinking in higher education: A review of the literature*. 2020.
- [LS07] A. Lewins and C. Silver. *Using software in qualitative research: A step-by-step guide*. 2007.
- [LSG⁺20] Y. Li, diSessa A. A. Schoenfeld, A. H., A. C. Graesser, L. C. Benson, L. D. English, and R. A. Duschl. On computational thinking and stem education. *Journal for STEM Education Research*, 3.2:147—166, 2020.
- [MKB99] S. Magnusson, J. Krajcik, and H. Borko. Nature, sources, and development of pedagogical content knowledge for science teaching. *Examining Pedagogical Content Knowledge: The Construct and its Implications for Science Education*, pages 95–132, 1999.
- [MSCD⁺10] J. Malyn-Smith, B. Coulter, J. Denner, I. Lee, J. Stilles, and L. Werner. Computational thinking in k-12: Defining the space. *Society for Information Technology Teacher Education International Conference*, 081:3479–3484, 2010.
- [NRC11] NRC. *REPORT OF A WORKSHOP ON THE PEDAGOGICAL ASPECTS OF COMPUTATIONAL THINKING*, pages 6–35. 2011.
- [Pap80] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. 1980.
- [PCWH20] Nadia Parsazadeh, Pei-Yu Cheng, Ting-Ting Wu, and Yueh-Min Huang. Integrating computational thinking concept into digital storytelling to improve learners’ motivation and performance. *Journal of Educational Computing Research*, 10 2020.
- [pro] Onderzoek project. <https://www.nwo.nl/projecten/40518540153-0>.
- [Rol06] C. Roland. Digital stories in the classroom. *School Art*. 7. 105 (7), page 26, 2006.
- [SBB17] V. Schmidt and R. Brandt-Bosma. *Passende perspectieven vmbo. Leerroutes rekenen voor vmbo-bb/kb*. 2017.
- [SC90] A. Strauss and J. M. Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. 1990.
- [SDMJ18] B. Sabitzer, H. Demarle-Meusel, and M. Jarnig. Computational thinking through modeling in language lessons. pages 1913–1919, 2018.
- [Shu86] L. S. Shulman. Those who understand: Knowledge growth in teaching. *Educational researcher*, pages 4–14, 1986.
- [SSSF14] C. Sneider, C. Stephenson, B. Schafer, and L. Flick. Computational thinking in high school science classrooms. *The Science Teacher*, 081, 2014.
- [SW13] C. Selby and J. Woollard. *Computational thinking: the developing definition. Project Report*. 2013.

- [Swa15] S. Swaid. Bringing computational thinking to stem education. volume 3, 2015.
- [TD16] M. Tedre and P. J. Denning. *The Long Quest for Computational Thinking*, pages 120–127. 2016.
- [TFVdH14] A. Thijs, P. Fisser, and M. Van der Hoeven. *21e eeuwse vaardigheden in het curriculum van het funderend onderwijs: een conceptueel kader*. 2014.
- [VFG⁺15] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav. Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20.4:715–728, 2015.
- [WGP19] K. P. Waterman, L. Goldsmith, and M. Pasquale. Integrating computational thinking into elementary science curriculum: an examination of activities that support students’ computational thinking in the service of disciplinary learning. *Integrating Computational Thinking into Elementary Science Curriculum: an Examination of Activities that Support Students’ Computational Thinking in the Service of Disciplinary Learning*, pages 53–64, 2019.
- [Win06] J. M. Wing. Computational thinking. *Communications of the ACM*, 49.3:33–35, 2006.
- [Yin84] R. K.. Yin. *Case Study Research: Design and Methods*. 1984.

7 Toestemmingsverklaring

TOESTEMMINGSVERKLARING

Naam onderzoek: Computational Thinking in Context: Een leerlijn voor PO en VO

Verantwoordelijke onderzoeker: Sabiha Yeni en Nataša Grgurina

Algemeen informatie over het onderzoek.

Sabiha Yeni (onderzoeker van Universiteit Leiden) en Nataša Grgurina (onderzoeker van Radboud Universiteit) doen onderzoek de naar het ontwikkelen van een leerlijn over de integratie van computational thinking in verschillende vakken in het PO en VO/

Doel en uitvoering van het onderzoek

Het doel van dit deelonderzoek is om de *pedagogical content knowledge* van leraren rondom het integreren van computational thinking in hun lespraktijk in kaart te brengen. We doen dat door met individuele leraren een gesprek van ongeveer een uur te voeren. Van deze gesprekken worden video- en geluidsopnames gemaakt

Risico's en ongemakken

Er zijn bij dit onderzoek geen risico's voor jullie gezondheid of jullie veiligheid.

Vertrouwelijkheid van de onderzoeksgegevens (

In dit onderzoek worden video-opnames en geluidsopnames gebruikt. We vragen geen naam van de docenten tijdens data verzameling. Als een docent de naam doorgeeft, verwijderen we jullie naam uit de gegevens. De geluidsopnames worden uitgeschreven. Video-opnames en geluidsopnames worden na het onderzoek vernietigd. De anonieme gegevens die we in dit onderzoek verzamelen, zullen door de onderzoekers gebruikt worden voor publicaties (artikelen/presentaties). De gegevens worden allemaal veilig bewaard voor tenminste 10 jaar volgens de richtlijnen van Radboud Universiteit. Alleen de betrokken onderzoekers kunnen de opgeslagen gegevens inzien.

Vrijwilligheid

De deelname van deze onderzoek is helemaal vrijwillig. Je kunt elk moment tijdens het onderzoek stoppen met meedoen. Al je gegevens worden dan definitief verwijderd.

Vragen over het onderzoek

Als je vragen hebt mag je deze altijd stellen aan je aan Sabiha Yeni en Nataša Grgurina. Je kunt ze mailen via sabihaakgun@gmail.com en natasa.grgurina@ru.nl

Het is voor ons heel belangrijk en fijn als je met het onderzoek mee wilt doen!

Met vriendelijke groet,

Sabiha Yeni en Nataša Grgurina



Verklaring deelnemer

Ik heb uitleg gekregen over het doel van het onderzoek. Ik heb vragen mogen stellen over het onderzoek. Ik neem vrijwillig aan het onderzoek deel. Ik begrijp dat ik op elk moment tijdens het onderzoek mag stoppen als ik dat wil. Ik begrijp hoe de gegevens van het onderzoek bewaard zullen worden en waarvoor ze gebruikt zullen worden. Ik stem in met deelname aan het onderzoek.

Naam: Geboortedatum:

Handtekening: Datum:

Verklaring uitvoerend onderzoeker

Ik verklaar dat de deelnemer juist heb geïnformeerd over het onderzoek en dat ik mij houd aan de richtlijnen voor onderzoekers.

Naam:

Handtekening: Datum:.....



Nijmegen, 30/9/2020

Geachte heer, mevrouw,

In het kader van het onderzoeksproject Computational thinking in context onderzoeken we op de Radboud Universiteit en op Leiden Universiteit hoe computational thinking in verschillende vakken op basisschool en in het voortgezet onderwijs geïntegreerd kan worden. Daarom hebben we samen met de docent van uw zoon/dochter lesmateriaal ontwikkeld dat binnenkort in de les gebruikt gaat worden.

Om na te gaan in hoeverre dit lesmateriaal bijdraagt aan het leren van leerlingen, krijgen de leerlingen tijdens het gebruik van dat lesmateriaal enkele keren een korte vragenlijst voorgelegd. Aanvullend zullen enkele leerlingen mondeling worden geïnterviewd om de vragenlijst te verduidelijken, en in dat geval worden er geluidsopnames gemaakt van die gesprekken. Zowel vragenlijsten als interviews worden gebruikt om het leerproces van de leerlingen beter te begrijpen en om hun mening over het lesmateriaal en de lessen te horen. In de lessen kunnen ook video-opnames gemaakt worden om te zien hoe er met het lesmateriaal wordt gewerkt.

Alle verzamelde data zullen veilig opgeslagen worden en geanonimiseerd verwerkt worden. In dit onderzoek worden er geen persoonsgegevens opgeslagen of verwerkt. De geluidsopnames worden uitgeschreven. De anonieme gegevens die we in dit onderzoek verzamelen, zullen door wetenschappers gebruikt worden voor publicaties (artikelen/presentaties). De gegevens worden allemaal veilig bewaard voor tenminste 10 jaar volgens de richtlijnen van Radboud Universiteit. Alleen de betrokken onderzoekers kunnen de opgeslagen gegevens inzien.

De deelname van deze onderzoek is helemaal vrijwillig. Uw kind kan elk moment tijdens het onderzoek stoppen met meedoen. Al zijn/haar gegevens worden dan definitief verwijderd.

Graag willen we uw toestemming voor deelname van uw zoon/dochter aan dit onderzoek vragen. Wilt aangeven of u wel of geen toestemming geeft door online formulier in te vullen door deze link (<https://tinyurl.com/CT-Toestemmingsverklaring>) te volgen?



Als u vragen hebt mag u deze altijd stellen aan de docent of aan de onderzoekers Nataša Grgurina en Sabiha Yeni.

Het is voor ons heel belangrijk en fijn als uw zoon/dochter met het onderzoek mee doet!

Met vriendelijke groet,

Nataša Grgurina
Radboud Universiteit, Bètavakdidactiek
Postbus 9010
6500 GL NIJMEGEN
natasa.grgurina@ru.nl

Sabiha Yeni
Leiden Universiteit, LIACS
Niels Bohrweg 1
2333 CA Leiden
s.yeni@liacs.leidenuniv.nl
