



**Universiteit  
Leiden**  
The Netherlands

**Opleiding Informatica & Economie**

Het gebruik van Sonic Pi door kinderen met een visuele beperking

Frederieke Hörchner

Begeleiders:

Anna van der Meulen & Felienne Hermans

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

[www.liacs.leidenuniv.nl](http://www.liacs.leidenuniv.nl)

14/12/2021

## Samenvatting

In dit onderzoek is gekeken naar hoe kinderen met een visuele beperking (voor het eerst) gebruik maken van het programma Sonic Pi. Dit is gedaan door een casus te bekijken waarin 2 visueel beperkte kinderen van 11 en 12 jaar oud voor het eerst met Sonic Pi werkten door middel van een opdracht onder begeleiding van een testleider. Daarnaast is er een semi-gestructureerd interview gehouden met de maker van Sonic Pi Sam Aaron. Hierin zijn vragen gesteld over leren programmeren met Sonic Pi en is er gepraat over de veranderingen van het programma om het toegankelijker te maken. Ook is er gekeken naar reeds onderzochte problemen bij programmeren met een visuele beperking om te kijken of dit overeen komt met de ervaring van de kinderen. Het is gebleken dat de kinderen tegen een aantal van de bekende problemen aanliepen, waaronder dat de screen reader niet goed werkt met het programma en het soms moeilijk is om de correcte plek in de code te vinden. Er zijn sinds de observatie een aantal nieuwe functionaliteiten aan Sonic Pi toegevoegd die onder andere deze problemen zouden moeten verhelpen, maar er moet nog onderzocht worden of deze de problemen daadwerkelijk verhelpen. De kinderen waren enthousiast over het programmeren in Sonic Pi en het lijkt dus of dit tekst based programma met muziek als output een geschikt programma is om kinderen met een visuele beperking te leren programmeren.

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>4</b>
<b>2</b>	<b>Theoretisch kader</b>	<b>4</b>
2.1	Sonic Pi	4
2.2	Programmeren met een visuele beperking	5
2.3	Kinderen met een visuele beperking	6
<b>3</b>	<b>Methode</b>	<b>8</b>
3.1	Participanten	8
3.2	Procedure	8
3.3	Materialen	8
3.4	Analyse	9
<b>4</b>	<b>Resultaten</b>	<b>11</b>
4.1	Casus	11
4.1.1	Samenvatting	11
4.1.2	Coderingen	12
4.2	Interview	12
4.2.1	Het ontstaan van Sonic Pi	12
4.2.2	Wat is er aan de toegankelijkheid gedaan?	13
4.2.3	Welke uitdagingen zijn er nog	13
<b>5</b>	<b>Discussie en Conclusie</b>	<b>14</b>
5.1	Discussie	14
5.2	Beperkingen	15
5.3	Conclusie	15
	<b>Referenties</b>	<b>17</b>

# 1 Inleiding

In Nederland zijn er naar schatting 378800 mensen met een visuele beperking.[KVI+11] Hierbij zijn de verschillende gradaties van blindheid niet uit elkaar gehaald. Wanneer je minder dan 5% kan zien en je gezichtsveld kleiner is dan 10 graden val je in de categorie “maatschappelijk blind”. Het kan zijn dat mensen die wetenschappelijk blind zijn nog licht en donker kunnen onderscheiden. Volledig blind is wanneer iemand ook geen licht meer ziet. Het wordt geschat dat er in Nederland ongeveer 2400 kinderen tussen de 0 en 14 jaar met een visuele beperking zijn.[KVI+11] De hoeveelheid kinderen met een visuele beperking wordt niet veel groter over de jaren in vergelijking met volwassenen. De meeste kinderen die een visuele beperking hebben zijn ermee geboren, maar bij volwassenen kan een visuele beperking ook het resultaat zijn van een ongeluk of ouderdom. De vergrijzing van de bevolking zorgt er ook voor dat het aandeel mensen met een visuele beperking steeds groter wordt onder volwassenen.

Een deel van de kinderen met een visuele beperking gaat naar het speciaal onderwijs, maar de meeste kinderen (ruim 80%) gaan niet naar het speciaal onderwijs[Vis], en volgen gewoon school met kinderen die kunnen zien; de leerkrachten en kinderen worden wel ondersteund door twee instituten die gespecialiseerd zijn in visuele beperkingen: Visio en Bartiméus. Dit kan goed zijn voor hun ontwikkeling, maar het probleem is dat een groot deel van de schoolmaterialen niet toegankelijk zijn voor kinderen met een visuele beperking. Om inclusief onderwijs te kunnen verzorgen moet het onderwijs en materiaal geschikt zijn voor zowel de kinderen in het speciaal onderwijs als in het reguliere onderwijs. Het veelvuldig gebruik van visuele uitleg of voorbeelden is een voorbeeld van iets wat ontoegankelijk is voor visueel beperkte kinderen. Met toegankelijkheid wordt er hier bedoeld dat het voor iemand met een visuele beperking goed en prettig te gebruiken is. Maar ook programmeren, waar deze scriptie over gaat, kan een uitdaging zijn. Coderen wordt steeds populairder en begint ook een prominentere rol in het onderwijs te spelen. De meeste codeer programma’s zijn niet ontwikkeld voor mensen met een visuele beperking en al helemaal niet voor kinderen die een visuele beperking hebben. Wat deze uitdagingen inhouden wordt later uitgelegd.

De focus van dit onderzoek zal liggen op kinderen die in groep 8 zitten en onderwijs volgen in het speciaal onderwijs. Dat is het einde van de basisschool. De kinderen hebben dan de leeftijd van 11 of 12 jaar. Hierbij kijken we naar het gebruik van Sonic Pi, een programma om muziek in te programmeren, met een screen reader. De onderzoeksvraag is dan ook:

*Wat zijn de ervaringen in het gebruik van Sonic Pi met een screen reader voor kinderen met een visuele beperking?*

## 2 Theoretisch kader

### 2.1 Sonic Pi

Sonic Pi is een muziek programmeertaal wat origineel geschreven was voor de Raspberry Pi. De Raspberry Pi is een kleine goedkope computer, ter grootte van een credit card [Pia]. Het is geen krachtige computer, maar wel goed genoeg om simpele programma’s op te draaien. De Raspberry Pi was net uitgekomen in 2012 en omdat er in het Verenigd Koninkrijk een nieuw curriculum werd gestart specifiek gericht op computer science op school werd een project gestart vanuit the

University of Cambridge om een programma te maken waarmee kinderen op een aantrekkelijke manier konden leren programmeren. Er zijn al verschillende scholen die Sonic Pi gebruiken om de basis van programmeren te leren[Pic], maar het is niet bekend hoeveel scholen dit op dit moment doen. Ze maken hierbij gebruik van de tutorial op de website van Sonic Pi of van de beschikbare plannen gemaakt door vrijwilligers. Maar Sonic Pi is niet alleen gemaakt om er mee te leren coderen, de code die de muziek als output heeft kan, met behulp van allerlei commando's en functies, zo complex worden als je maar kunt bedenken, en hierdoor is het geschikt voor alle leeftijden. Eén van de mogelijkheden van Sonic Pi is het gebruiken als live instrument. Door de code live aan te passen wordt het dynamisch voor een optreden.

Om Sonic Pi toegankelijk te maken, zijn er een aantal bewuste keuzes gemaakt. Zo is, in tegenstelling tot veel programma's, het design simpel en duidelijk met grote knoppen en vriendelijke kleuren. Op deze manier ziet het er niet eng uit en is de drempel om te beginnen met coderen niet zo hoog[Pic]. Veel programma's die gericht zijn op beginners zijn block based en dus ontoegankelijk voor mensen met een visuele beperking[MVT+20]. Sonic Pi is daarentegen tekst based. Tekst based ziet er in eerste instantie meteen lastiger uit, maar het is juist ontworpen om simpel te zijn. Een simpele melodie is al te maken met maar een paar regels code met begrippen die makkelijk uit te leggen zijn aan kinderen of beginners. Ook is het programma erg licht in gebruik en geschikt voor zowel Linux, Windows als MacOS. Dit zorgt ervoor dat je geen dure computer nodig hebt om gebruik te kunnen maken van Sonic Pi en ook dit verlaagt de drempel. Daarnaast is het ook nog gratis[Pib].

## 2.2 Programmeren met een visuele beperking

Als het gaat om programmeren door individuen met een visuele beperking, is bekend dat er problemen zijn. Zo is er een onderzoek gedaan onder blinde ontwikkelaars om uitdagingen te vinden[AL16]. Hieruit is gebleken dat IDE's een probleem kunnen vormen. Een IDE, oftewel Integrated Development Environment, is een stuk software met een editor (de plek om de code te typen), compiler (een programma wat de geschreven code omzet in de de doeltaal en gebruikt kan worden als output), debugger (hulpmiddel om fouten op te sporen) en eventueel meer functionaliteiten in één. Sonic Pi is in dit opzicht vergelijkbaar met een IDE, omdat Sonic Pi ook al deze functionaliteiten in één programma heeft. Het is dan dus ook aannemelijk dat het gebruik van Sonic Pi vergelijkbare problemen met zich mee brengt als het gebruik van IDE's. Het is de IDE in combinatie met screen readers wat voor problemen zorgt. Een screen reader is een ondersteunende technologie welke voornamelijk door blinden en slechtzienden wordt gebruikt om tekst van een scherm te kunnen lezen. Dit programma kan tekst voorlezen of omzetten in een ander formaat wat toegankelijk is (denk hierbij aan bijvoorbeeld braille). Een screen reader is niet programma gebonden. Het is namelijk een applicatie voor op bijvoorbeeld een computer of telefoon, die de pagina's van andere programma's voorleest. Een programma kan wel in meer of mindere mate toegankelijk zijn voor een screen reader. Hoe veel het programma aansluit bij de screen reader is van groot belang. Zo is het belangrijk dat het voor de gebruiker duidelijk is waar op een pagina diegene zich bevindt en dat er geen onderdelen worden overgeslagen. Wanneer er bijvoorbeeld een afbeelding is moet deze een duidelijke omschrijving hebben. Hierdoor weet degene wat er gebeurt en is er geen missende informatie.

Een van de frustraties is dus de IDE die niet goed aansluit bij de screen reader. Hierdoor kunnen blinde mensen geneigd zijn om een tekst editor te gebruiken in plaats van een IDE[ALH17]. De IDE kan te ingewikkeld zijn of te veel afhankelijk van visuele elementen om te navigeren/gebruiken of helemaal niet compatibel met een screen reader. Hoewel veel IDE's gebruik maken van visuele aspecten, die zeer nuttig zijn voor mensen die daar baat bij hebben, maar er is altijd een manier om iets op een aangepaste of compleet andere manier te doen[KL16]. Soms is de IDE zelf niet het probleem maar de documentatie of bekendheid. Het kan dus zijn dat er een feature wel in het programma zit, maar dat de gebruiker niet van het bestaan van de functie weet en hierdoor dus deze gemakken misloopt. De manier waarop een programma of IDE aan de mens wordt gebracht is dus erg belangrijk. En (duidelijke) updates ook.

Er is ook een onderzoek gedaan waarbij frustraties van mensen die een screen reader op het web gebruikte werden onderzocht[LAKM07]. Hoewel het web natuurlijk een breed begrip is, zijn er frustraties die niet alleen op het web van toepassing zijn. Een opvallend vaak voorkomende frustratie was dat de pagina layout, bijvoorbeeld verschillende kolommen, voor verwarring zorgde bij de output van de screen reader. Iemand met een goede visus kan door een tekst scannen en een combinatie van de onderdelen van de opmaak gebruiken om een gewenst onderdeel te vinden, waar iemand met een screen reader de tekst regel voor regel doorneemt. Een layout is niet overal gelijk en daar kunnen niet alle screen readers goed mee omgaan. Het is dus belangrijk dat een pagina duidelijk te begrijpen is.

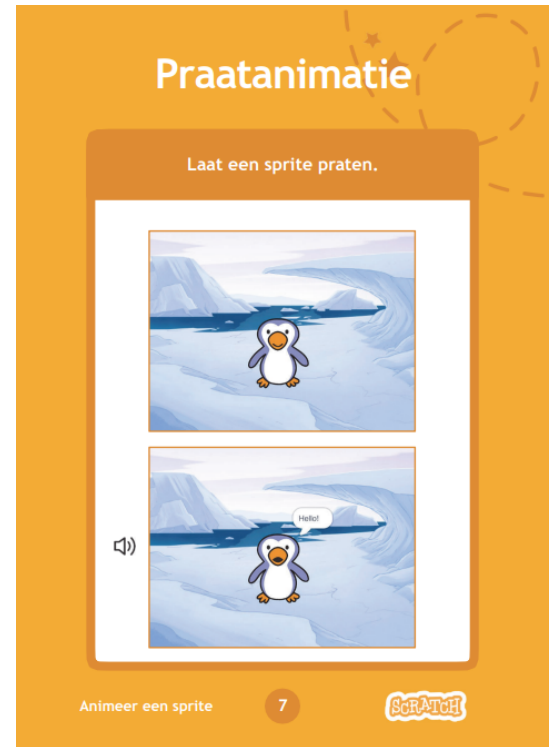
Een uitdaging die hierop aansluit is overzicht houden over een stuk geschreven code wanneer je deze niet kunt zien. Bij een doorlopend stukje code is het nog wel te overzien, maar hoe complexer de code, hoe lastiger het is om overzicht te houden. Dit kan er ook voor zorgen dat de snelheid van het programmeren beperkt is. Vooral het navigeren van de code is hierbij een uitdaging. Wanneer je bezig bent met programmeren werk je aan een stukje code binnen een groter stuk code. De context van dat stuk is van groot belang. Wanneer je geen visuele beperking hebt kun je door code scrollen en de context voor je zien, dit is moeilijker wanneer je afhankelijk bent van een screen reader. Er zijn tools om een duidelijker overzicht te krijgen van een hiërarchische structuur van sommige talen[BML15], deze zijn te gebruiken naast een editor.

## 2.3 Kinderen met een visuele beperking

Bovenstaande inzichten zijn vooral gebaseerd op onderzoek onder volwassenen met een visuele beperking, maar de focus van dit onderzoek ligt op kinderen. Wanneer kinderen leren programmeren zullen ze snel in aanraking komen met een block based programma zoals Scratch. Hier volgt een voorbeeld van een scratch programma:



(a) Code



(b) Output

Figuur 1: Scratch voorbeeld [Scr]

Met Scratch kunnen er door het slepen van blokken verhalen, spellen en animaties gecreëerd worden. De bovenstaande afbeelding is een voorbeeld van de samengevoegde blokken en de animatie die het als output geeft. Om gebruik te maken van block based programmeren hoeven ze geen ingewikkelde syntax te kennen, omdat de vormen en kleuren een indicatie geven van wat ermee gedaan kan worden [RMMH<sup>+</sup>09]. Maar die vormen zijn voor kinderen met een visuele beperking niet bereikbaar. Ze kunnen weinig of niet gebruik maken van een muis en dus is het blokken slepen en voorlezen door een screen reader onhandig of zelfs onmogelijk.

Het intuïtieve deel wat het slepen en de kleuren van block based programma's brengt is voor kinderen met een visuele beperking al vervallen en daarvoor in de plaats komt met Sonic Pi, een tekst based programma, een eenvoudige syntax die voor deze kinderen wel toegankelijk is met een screen reader. Doordat de output een stuk muziek is, is deze toegankelijk zowel met als zonder visuele beperking en tevens maakt dit het aantrekkelijk voor kinderen. Tekst based programmeren brengt bij ervaren volwassenen problemen met zich mee en het is dus te verwachten dat dit ook zo zal zijn voor kinderen die gebruik maken van Sonic Pi. Daarom is het belangrijk dat het onderzocht wordt of dit bij Sonic Pi ook het geval is.

## 3 Methode

Het uitgevoerde onderzoek is een kwalitatief onderzoek en bestaat uit twee onderdelen. Er is een casus bestudeerd van twee blinde kinderen die een opdracht aan het doen waren met Sonic Pi. Daarnaast is een interview gehouden met de maker van Sonic Pi.

### 3.1 Participanten

De participanten van het eerste deel van het onderzoek waren twee meisjes van 11 en 12 jaar oud die voor het eerst met Sonic Pi werkten. Het ene meisje is helemaal blind en het andere meisje kan een heel klein beetje zien als ze heel erg dicht op het scherm zit, maar ook zij is wetenschappelijk gezien blind. Zo hebben ze samen, onder begeleiding, een opdracht gedaan. Het is voor deze meiden niet mogelijk om deel te nemen aan het reguliere onderwijs en daarom zitten ze op het speciaal onderwijs. Deze school bevindt zich in een wat kleinere plaats in een minder dicht bevolkt deel van Nederland. De participant van het tweede deel was Sam Aaron. Sam is een computer scientist en heeft een PhD gedaan in het ontwerpen van Domain Specific Languages[Aar08]. Hij is tevens de maker van Sonic Pi. Doordat hij zelf degene is die aan Sonic Pi werkt is hij degene die het meest kon vertellen over het programma en al geïmplementeerde of mogelijke features.

### 3.2 Procedure

Het eerste deel van het onderzoek was een al bestaande video bestuderen. Deze casus was onderdeel van een groter onderzoek wat in het najaar van 2019 is uitgevoerd. Daarin hebben verschillende blinde en slechtziende kinderen opdrachten gedaan met allerlei materialen waaronder een opdracht (waar ze ongeveer 45 minuten de tijd voor hadden) met Sonic Pi en ze zijn gefilmd tijdens dit proces. Het doel van dit onderzoek was inzicht krijgen in de toegankelijkheid van deze verschillende materialen voor kinderen met een visuele beperking. Er was bij dit onderzoek toestemming van de ouders verkregen. De casus die voor dit onderzoek gebruikt was, was onderdeel van de derde les van het hierboven genoemde grotere onderzoek.

Het tweede deel van het onderzoek was een interview. Er is contact opgenomen met Sam Aaron. Hij is zelf momenteel bezig met aanpassingen aan Sonic Pi maken die onder andere invloed hebben op het gebruik door iemand met een visuele beperking. Het interview heeft plaatsgevonden via Zoom en was in het Engels. Zowel Sam als de interviewer hadden hun webcam aan, wat zorgde voor een natuurlijk gesprek, maar alleen het geluid is opgenomen. Het was een semi-gestructureerd interview, met vragen die waren voortgekomen uit gelezen literatuur, de bestudeerde video en de updates van Sonic Pi zelf. Er is gekozen voor vooraf opgestelde vragen omdat het duidelijk was wat voor informatie we zochten, maar tijdens het interview was er de ruimte om door te vragen naar een specifiek deel van het antwoord. Door het op deze manier te doen, en niet via een vragenlijst, kan er een gesprek worden aangegaan en kunnen antwoorden en meningen worden uitgediept.

### 3.3 Materialen

De tool die onderzocht is is het eerder al genoemde Sonic Pi. In het onderzoek van de bestudeerde casus hebben de kinderen gebruik gemaakt van de toen meest recente versie van Sonic Pi: 3.1.0 -



'Sauna'. Sindsdien is er op 28 februari 2020 een nieuwe versie uitgebracht. V3.2.0 – 'Tau', dit was een update die zich deels richtte op toegankelijkheid. Het is dus belangrijk om rekening te houden met het feit dat de features die bij die update zijn toegevoegd nog niet gebruikt zijn tijdens het onderzoek van de bestudeerde video.

Ook is het belangrijk om te melden dat doordat de kinderen die in de casus met Sonic Pi gewerkt hebben geen eerdere ervaring met Sonic Pi of andere programmeertalen hadden. De twee kinderen hebben dus samen een opdracht gemaakt die gericht was op beginners. De opdracht bestond uit: noten van verschillende hoogtes leren afspelen, pauzes van verschillende lengtes toevoegen en het instrument veranderen. Sonic Pi heeft meer mogelijkheden dan deze opdracht laat zien, maar dit was niet mogelijk voor beginners in de geplande tijd.

### 3.4 Analyse

De bestudeerde casus is voor dit onderzoek gecodeerd en getranscribeerd in een van te voren opgesteld schema met gedragscategorieën met daarbinnen concreet identificeerbare gedragingen. In totaal waren er 17 categorieën, waarvan 10 voor dit onderzoek gebruikt zijn (de overige categorieën hadden meer specifiek betrekking op computational practices van de kinderen, wat buiten de focus valt). De gedragscategorieën waren:

1. Zelfstandig toegang tot materiaal
  - Volgen visueel
  - Volgen auditief
2. Met hulpmiddel toegang tot materiaal
  - Screenreader geeft toegang
  - Zoomfunctie geeft toegang
3. Via ander toegang tot materiaal
  - Krijgen uitleg testleider
  - Tactiel begeleiding door ander kind
4. Leren werken met materiaal
  - Exploreren door uitproberen acties
  - Luisteren naar instructie
5. Aan opdracht werken met materiaal
  - Werken aan opdracht
  - Bezig met iets anders dan materiaal

## 6. Samenwerking aanwezigheid

- Advies vragen aan elkaar
- Coördineren tussen kinderen

## 7. Samenwerking afwezigheid

- Aan tegengestelde doelen werken
- Competitie aangaan

## 8. Creativiteit en exploratie

- Nieuwe stap toevoegen binnen gebruikte onderdelen
- Werken aan nieuw idee opdracht

## 9. Positieve ervaring materiaal

- Lachen
- Opgewonden bewegen

## 10. Negatieve ervaring

- Negatief uitspreken
- Verwarring/ervaren moeilijkheid laten zien

Dit leidde tot een ingevuld schema, waarin alle verbale en non-verbale gedragingen volledig aangeduid waren. Dit schema is vervolgens gebruikt voor het schrijven van een precieze samenvatting die het verloop en de verschillende relevante gedragingen, binnen de 10 categorieën die hierboven zijn genoemd, van de kinderen tijdens het werken met Sonic Pi aanduidt. Aan de hand van de gedragingen en transcripties is er een samenvatting gemaakt van gedrag en ervaring van de kinderen tijdens de hele sessie.

Het gehouden interview is getranscribeerd en daarna zijn de relevante informatie gemarkeerd en eruit gehaald. Kleine verspreking zijn wel aangepast tijdens het transcriberen, maar incorrecte zinsstructuren zijn niet verbeterd. Vervolgens is informatie die direct gerelateerd was aan onderzoeksvraag uit het interview gehaald en gesorteerd per onderwerp.

1. Het ontstaan van Sonic pi
2. De toegankelijkheid van Sonic Pi
3. De blijvende uitdagingen

Ook andere onverwachte maar relevante informatie uit het interview werd genoteerd. Deze informatie is niet allemaal relevant voor dit onderzoek, maar kan wel interessant zijn voor eventuele vervolgonderzoeken.

## 4 Resultaten

In dit stuk worden de resultaten besproken. Ook hier zal het verdeeld worden over 2 delen: De bestudeerde casus van blinde kinderen die werken met Sonic Pi en het interview met Sam Aaron.

### 4.1 Casus

#### 4.1.1 Samenvatting

De casus is compleet gecodeerd en daarna in woorden samengevat. Hier volgt deze samenvatting:

De kinderen zitten samen met de testleider achter de computer en ze krijgen van de testleider een introductie over Sonic Pi en de opdracht die ze gaan doen. Ze krijgen een uitleg over de inhoud (muziek programmeren) en over de werkwijze (omdat Sonic Pi niet helemaal werkt met een screen reader zullen ze code heen en weer kopiëren met behulp van Word). Dan starten ze de opdracht en leren ze dat je een noot moet typen met “play”. Ze leren dat hoe groter het getal is, hoe hoger de noot is waarna ze een noot kiezen. Daarna leren ze de toetsenbord combinatie voor het afspelen van en wordt de noot afgespeeld en besproken. Hierna worden er nog wat noten toegevoegd. Ze komen erachter dat de noten allemaal tegelijk worden afgespeeld.

De kinderen krijgen uitleg over dat het mogelijk is om een pauze tussen de noten toe te voegen met “sleep” en het aantal seconden. Een noot wordt weggehaald en een andere wordt toegevoegd en nu hebben ze twee noten met een pauze ertussen. Deze code wordt weer van Word naar Sonic Pi gekopieerd en afgespeeld. De code werkt en ze spelen wat met de lengte van de pauze tussen de noten waarbij ze typen in Sonic Pi, hierbij is er even een korte verwarring over ingetypte code, maar uiteindelijk staat het correcte stukje code in Sonic Pi. Door te proberen komen ze erachter dat te lage noten niet hoorbaar zijn. Na een tijdje hebben ze een stukje muziek van een aantal noten met pauzes ertussen.

Hierna begint de testleider een laatste uitleg over wat er nog meer bij muziek hoort (en waar ze het al eerder over hebben gehad): instrumenten. De testleider legt uit hoe je een instrument kan kiezen en vertelt de opties. De kinderen wisselen een paar keer van instrument en spelen tussendoor de code af en besluiten dat ze de originele het best vinden. Ze besluiten nog wel wat aanpassingen te willen doen aan de toonhoogtes en de pauzes en dit lukt met wat hulp van de testleider om de goede plek in de code aan te wijzen. Ze blijven wat overleggen over code en besluiten de noten van laag naar hoog te willen doen. Ook hier hebben ze weer een beetje hulp bij nodig. In overleg met de testleider willen ze ook nog noten van hoog naar laag toevoegen, wat hen zonder hulp lukt. Ze spelen dit alles af en zijn blij met het resultaat.

Daarna zegt de testleider dat ze nu een afsluiting kunnen maken. Er is niet genoeg tijd meer om veel aan het nummer toe te voegen, dus als afsluiten voegen ze een paar noten zonder pauze ertussen toe. Ook dit lukt zonder hulp van de testleider. Ze spelen het nummer een paar keer af en zijn tevreden. En met hulp van de testleider slaan ze het project op.

### 4.1.2 Coderingen

De verschillende gedragingen uit de casus zijn ook gecodeerd in een schema. Hieronder volgt een samenvatting van de frequentie hiervan:

De eerste 3 categorieën hebben betrekking op het gebruiken van het materiaal. Zo laat de eerste categorie, zelfstandig toegang, zien dat deze toegang visueel, auditief of tactiel was. Het grootste gedeelte van de tijd was dit auditief, en tactiel kwam maar een enkele keer voor. De tweede categorie, met hulpmiddel toegang tot materiaal, laat zien dat door de hele opdracht heen de screen reader regelmatig (maar niet altijd) gebruikt werd. Over het algemeen gaf de screen reader toegang, maar af en toe ook niet. Vervolgens laat de derde categorie, via ander toegang tot materiaal, zien dat ze geholpen worden door de testleider, maar ook elkaar een enkele keer helpen door middel van tactiele begeleiding. De twee categorieën die hierop volgen hebben beide betrekking op het werken aan de opdracht. In de vierde categorie, leren werken met materiaal, is te zien dat ze vooral gebruik maken van de uitleg van de testleider, maar ook een enkele keer zelf dingen proberen. Daarnaast laat de vijfde categorie, aan opdracht werken met materiaal, zien dat ze het grootste deel van de tijd werken aan de opdracht en maar een enkele keer even met iets anders bezig zijn dan het materiaal. De drie categorieën hierna gaan over de samenwerking tijdens de opdracht. Zo toont de zesde categorie, samenwerking aanwezigheid, dat de kinderen regelmatig elkaar om advies vragen en acties coördineren. En vervolgens laat de zevende categorie, samenwerking afwezigheid, zien dat ze maar een enkele keer niet goed samenwerken. Verder laat de achtste categorie, creativiteit en exploratie, zien dat ze maar een enkele keer zelf iets doen wat niet expliciet als taak in de opdracht staat. De laatste twee categorieën gaan over hoe de kinderen het materiaal hebben ervaren. Zo is in de negende categorie, positieve ervaring materiaal, te zien dat ze beide regelmatig lachen, glimlachen, opgewonden bewegen, en zich positief uitspreken over het materiaal. En tenslotte laat de tiende categorie, negatieve ervaring, een enkele keer zien dat ze een aantal keer verwarring/ervaren moeilijkheid laten zien, en zich ook een enkele keer negatief uitspreken over het materiaal.

## 4.2 Interview

### 4.2.1 Het ontstaan van Sonic Pi

Bij het ontstaan van Sonic Pi speelde samen dat Sam Aaron de inspiratie kreeg om muziek te coderen in plaats van te spelen met een instrument, en de behoefte aan een manier om kinderen geïnteresseerd te krijgen in coderen. De inspiratie om muziek te coderen ontstond op een technische conferentie bij een bar waar sprekers muziek aan het maken waren. Hij wilde op een programmeer conferentie de muziek kunnen coderen in plaats van een instrument zoals een gitaar te bespelen. Daar kwam samen met een vriend van hem, Jeff Rose, een systeem genaamd Overtone uit voort. Dat systeem is nooit erg groot geworden, omdat het te ingewikkeld was om aan mensen uit te leggen hoe het te gebruiken was. Een tijd later was hij onderzoeker bij Cambridge university. Hier waren ze op zoek naar iemand die een project kon maken om kinderen geïnteresseerd te krijgen in coderen. Dat was namelijk het moment dat er een nieuw curriculum voor informatica op scholen in het VK kwam, maar er was nog geen materiaal om de kinderen op een leuke manier te laten beginnen en zeker niet iets wat geschikt was voor iets zoals de Raspberry Pi (wat toentertijd een computer met een zeer laag vermogen was), welke net was uitgekomen. Sam Aaron stelde zichzelf voor met het idee om een simpele versie te maken van Overtone. Zo heeft hij zijn kennis over

domain specific languages (hij heeft een PhD in language design en in het speciaal domain specific languages), muziek en educatie gecombineerd en heeft samen met een docent een eerste versie gemaakt binnen twee weken. Daarbij hebben ze een les plan gemaakt en door dat uit te voeren zijn er steeds aanpassingen gedaan. Dit was een project van drie maanden. Tegenwoordig is het niet meer gefinancierd door een externe organisatie, maar werkt Sam Aaron zelf aan het programma.

#### **4.2.2 Wat is er aan de toegankelijkheid gedaan?**

Sonic Pi is niet gebouwd met toegankelijkheid in gedachte. Zo was het grote tekst blok waarin de code staat voorheen niet toegankelijk voor screen readers. Dat was een onderdeel wat Sam Aaron zelf niet geschreven had en het was dus niet mogelijk voor hem om dat zelf aan te passen. Degene die verantwoordelijk is voor de editor van Sonic Pi heeft het toegankelijk gemaakt en dit gaf kans om op meer verbeteringen te focussen.

Sam Aaron heeft, door overleggen met visueel beperkte mensen in de community en door zelf Sonic Pi proberen te gebruiken met een screen reader, sneltoetsen toegevoegd waarmee Sonic Pi kan worden genavigeerd en er heen en weer kan worden gesprongen tussen plekken. Ook is er in de hoek een venster toegevoegd welke verteld op welke lijn en welke regel je je bevind met de cursor. Daarnaast is er een high contrast mode toegevoegd. Dit is vooral handig voor mensen die nog wat zicht over hebben. Zwart op wit is makkelijker te onderscheiden dan licht of donker grijs.

Sommige van de bekende moeilijkheden zijn maar gedeeltelijk opgelost. Zo waren de voorkeursinstellingen, wat je als gebruiker zelf kunt instellen over beeld en geluid (zoals de volumeregelaar), niet toegankelijk. Sam Aaron wist niet waar het probleem lag en heeft in plaats van zijn tijd en energie te stoppen in het zoeken naar de oorzaak, ervoor gekozen om deze instellingen te kopiëren naar een andere plek die wel toegankelijk is. Zo is het probleem opgelost, maar staan de instellingen er wel dubbel. Deze manier van oplossen gebruikt hij ook voor de toegankelijkheid met een muis. Hij vertelde dat er zeker nog dingen zijn die verbeterd kunnen worden, maar dat er zoveel mogelijk nu gedaan kan worden met shortcuts of op plekken waarvan bekend is dat het toegankelijk is zonder het gebruik van een muis.

#### **4.2.3 Welke uitdagingen zijn er nog**

Sam Aaron benadrukte dat zorgen voor toegankelijkheid op het programma blijft staan. Omdat hij zelf geen visuele beperking heeft, zal hij dingen aan moeten passen wanneer gebruikers tegen moeilijkheden aanlopen. Hij vindt het geen probleem om dingen aan te passen voor bestaande gebruikers, in plaats daarvan focust hij zich meer op hoe makkelijk het is om te leren en hoe toegankelijk het is voor nieuwe gebruikers. Op die manier zal Sonic Pi beter worden voor iedereen.

Daarbij is het wel van belang of dat een onderdeel is wat hij zelf aan kan passen of niet. De open source delen kan hij, met genoeg tijd, geld en expertise, zelf aanpassen. Maar er zijn ook delen waar problemen liggen die closed source zijn. Zoals voorheen het tekst blok wat niet toegankelijk was. Zo ziet Sam Aaron ook problemen die zich voordoen met de screen reader. In de afgelopen jaren is er in tekst editors syntax highlighting toegevoegd om de verschillende delen van de code met kleuren te onderscheiden. Zoiets is er nog niet voor screen readers, want die leest code voor alsof het een

verhaal is. Hij zei: “So what would be the equivalent of that for screen readers? That’ll be interesting. That control to be able to nuance the screen reader, maybe to do it in different accents, or to read at different speeds or different volumes. I don’t know what it would be, but to have that control.”

Als laatste vertelde hij dat een uitdaging van het toegankelijk maken vooral de ervaring is. Er zijn niet genoeg mensen die weten hoe je toegankelijke software maakt. Computer science vakken zouden dit als onderdeel op moeten nemen en er moet bij software projecten vanaf de design requirements fase rekening mee gehouden worden. Hier is een fundamentele verandering voor nodig.

## 5 Discussie en Conclusie

### 5.1 Discussie

Het doel van dit onderzoek was het in kaart brengen van hoe kinderen met een visuele beperking gebruik maken van Sonic Pi. Dit is gedaan door een casus van twee kinderen die met Sonic Pi werken te bestuderen en een interview te houden met de maker van Sonic Pi.

In de casus zien we de kinderen aan het werk in Sonic Pi. Ze maken gebruik van de tekst editor Word om de code in te typen en deze dan naar Sonic Pi te kopiëren. Zoals uit verschillende onderzoeken is gebleken is het veel voorkomend dat programma’s niet goed werkt met een screen reader[ALH17]. Er is een moment dat een van de kinderen opmerkt dat het mogelijk is het tekstveld in Sonic Pi te gebruiken en typt hier ook iets, maar even later gebruiken ze weer Word. Dit is mede doordat de versie die gebruikt is tijdens het onderzoek niet de huidige nieuwste versie is. In het interview kwam naar boven dat het tekstveld nu toegankelijk is voor screen readers. Hierdoor zou het voor mensen met een visuele beperking niet nodig moeten zijn om een andere tekst editor te gebruiken, wat veel gebruiksvriendelijker is. De nieuwste versie moet ook helpen met de correcte plek in de code vinden. De kinderen waren een aantal keer op zoek naar locatie in de code, dus een scherm welke aangeeft waar de cursor zich bevindt kan dit probleem verhelpen. Deze functionaliteiten moeten nog onderzocht worden om te kijken of ze het beoogde resultaat hebben.

Door het toegankelijk maken van het tekst veld is het niet alleen mogelijk om een screen reader te gebruiken, maar er kan door mensen die slechtziend zijn (en nog overgebleven zicht hebben) gebruik gemaakt worden van de verschillend gekleurde en dik gedrukte woorden die de syntax weergeven. Zoals ook door Sam Aaron opgemerkt tijdens het interview is er helaas geen soortgelijke weergave met een screen reader. Een manier om syntax aan te geven met een screen reader, door bijvoorbeeld intonatie of tempo, zou programmeren, zeker voor beginners, denk ik een stuk vergemakkelijken. Het zou interessant zijn om te onderzoeken wat de beste manier is om syntax aan te geven met een screen reader.

Uit onderzoek is gebleken dat een frustratie van visueel beperkte mensen die gebruik maakt van een IDE niet alleen de screen reader compatibiliteit is, maar ook andere onderdelen welke anders vooral visueel zijn[ALH17]. Hierbij kunnen shortcuts een uitkomst bieden, maar dan moeten deze wel goed gedocumenteerd zijn. Wanneer iemand niet weet van het bestaan van een functionaliteit kan die deze ook niet gebruiken. Dit is ook naar voren gekomen in de casus. Er waren

al een aantal shortcuts waar de kinderen gebruik van maakten, zoals de shortcut om de code af te laten spelen. In de nieuwste versie van Sonic Pi zijn er nog meer shortcuts toegevoegd. Deze shortcuts zijn te vinden in Sonic Pi zelf in tekst bij de knoppen, maar staan ook op de website gedocumenteerd. De kinderen hebben met behulp van de testleider hun stuk code opgeslagen, maar daar is ook een shortcut voor, welke dus niet bekend was. Als er dus externe documentatie is kan dit helpen met het leren van Sonic Pi met minder tot geen hulp. En wanneer er wel hulp bij is, is de documentatie belangrijk om meteen functionaliteiten aan te kunnen leren. Ook deze nieuwe functionaliteit is nog niet onderzocht.

Hoewel de meeste programma's waar kinderen mee leren programmeren block based en niet tekst based zijn, lijkt Sonic Pi in de casus relatief intuïtief voor de kinderen. De kinderen werden enthousiast van de muziek en dat maakte de uitleg en start gemakkelijk. Sonic Pi heeft veel mogelijkheden en een stuk muziek kan zo moeilijk gemaakt worden als iemand zelf wil, dus ook erg simpel. Deze opties maken het een goed programma om in te leren programmeren. Dit lijkt ook zeker zo te zijn voor de kinderen. Er zijn al mensen bezig geweest met lesplannen, maar als deze aangepast zouden worden op de nieuwe versie en toegankelijk is voor kinderen met een visuele beperking (door visuele elementen niet noodzakelijk te maken) kan dit een goed programma zijn om kinderen, zowel met als zonder visuele beperking, te leren programmeren.

## 5.2 Beperkingen

Dit onderzoek heeft een aantal te benoemen beperkingen. Zo is dit onderzoek gebaseerd op één casus met twee leerlingen. Dit is een erg kleine groep en het kan dus zijn dat er zich andere problemen voor doen bij meer gebruikers. Het zou dus goed kunnen zijn om een onderzoek uit te voeren op een grotere schaal. Zo kan er gekeken worden of visueel beperkte kinderen grotendeels tegen dezelfde problemen aanlopen of dat er zich problemen voordoen waar we nog niet vanaf weten. Daarnaast is Sonic Pi geüpdatet na het onderzoek. Een deel van de bevindingen uit de casus kunnen hierdoor afwijken van de huidige ervaringen. Dit kan dus ook nog verder onderzocht worden om zo te kijken welke aanpassingen er nog meer voordelig zouden zijn.

## 5.3 Conclusie

Met dit onderzoek is gebleken dat de ervaringen van de kinderen die Sonic Pi gebruiken over het algemeen positief zijn. De problemen waar ze tegenaan lopen lijken veelal te zijn of kunnen worden verholpen door middel van updates van het programma. Zo was het tekstveld niet toegankelijk, maar is dit aangepast door middel van updates. Er is nog wel meer onderzoek nodig op dit gebied, want er is nog geen grootschalig onderzoek gedaan met de nieuwste update van Sonic Pi. Hierdoor kan er nog verder onderzocht worden of Sonic Pi echt een geschikt programma is om kinderen met een visuele beperking mee te leren programmeren.

## Referenties

- [Aar08] S. Aaron. A domain specific language for dynamic interest management within virtual environments. 2008.
- [AL16] Khaled Albusays and Stephanie Ludi. Eliciting programming challenges faced by developers with visual impairments: Exploratory study. In *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 82–85, 2016.
- [ALH17] Khaled Albusays, Stephanie Ludi, and Matt Huenerfauth. Interviews and observation of blind software developers at work to understand code navigation challenges. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, page 91–100, New York, NY, USA, 2017. Association for Computing Machinery.
- [BML15] C. M. Baker, Lauren R. Milne, and R. Ladner. Structjumper: A tool to help blind programmers navigate and understand the structure of code. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.
- [KL16] Varsha Koushik and Clayton Lewis. An accessible blocks language: Work in progress. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, page 317–318, New York, NY, USA, 2016. Association for Computing Machinery.
- [KVI<sup>+</sup>11] Jan EE Keunen, C Anton Verezen, Saskia M Imhof, GHMB Van Rens, MB Asselbergs, and JJ Limburg. Toename in de vraag naar oogzorg in nederland 2010-2020. *Ned Tijdschr Geneeskd*, 155(A3461):1–6, 2011.
- [LAKM07] Jonathan Lazar, Aaron Allen, Jason Kleinman, and Chris Malarkey. What frustrates screen reader users on the web: A study of 100 blind users. *International Journal of Human-Computer Interaction*, 22(3):247–269, 2007.
- [MVT<sup>+</sup>20] Cecily Morrison, Nicolas Villar, Anja Thieme, Zahra Ashktorab, Eloise Taysom, Oscar Salandin, Daniel Cletheroe, Greg Saul, Alan F Blackwell, Darren Edge, et al. Torino: A tangible programming language inclusive of children with visual disabilities. *Human-Computer Interaction*, 35(3):191–239, 2020.
- [Pia] Raspberry Pi. About us. <https://www.raspberrypi.org/about/>.
- [Pib] Sonic Pi. About us. <https://sonic-pi.net/>.
- [Pic] Sonic Pi. Live coding education. <https://sonic-pi.net/files/articles/Live-Coding-Education.pdf>.
- [RMMH<sup>+</sup>09] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, November 2009.



- [Scr] Scratch. Animeer een sprite. <https://resources.scratch.mit.edu/www/cards/nl/animation-cards.pdf>.
- [Vis] Visio. Onderwijs. <https://www.visio.org/nl-nl/onderwijs>.