# Def_InZicht 1718

**Uitgewerkte projectaanvraag InZicht 13e ronde**

## Algemene gegevens / General Information

Inclusive programming education: The accessibility of existing programming materials for visually impaired children

## Geplande duur / Planned duration

24 maanden / months

## Geplande startdatum / Planned start date

September 2018

## Samenvatting (maximaal 4159/5000 karakters)


Inclusief programmeeronderwijs: De toegankelijkheid van bestaand materiaal voor blinde en slechtziende leerlingen

De wereld om ons heen verandert razendsnel door technologische ontwikkelingen. Het onderwijs sluit hierbij aan en er is een breed gedragen consensus dat kinderen digitaal geletterd moeten worden. De opkomst van 'Computational Thinking' is nu zo concreet dat er leerlijnen worden geïmplementeerd die leerlingen leren programmeren. Hoewel dit een enorme (internationale) beweging is, is er nog nauwelijks gekeken hoe toegankelijk en bruikbaar materiaal voor blinde en slechtziende leerlingen is. Het is essentieel om hier tijdig op in te springen, juist omdat de bestaande materialen en curricula zeer visueel georiënteerd zijn. Toegang tot dit curriculum bereidt hen voor op een toekomst waarin communicatie met technologie cruciaal is. In dit project wordt inclusiviteit van bestaande programmeerlessen geanalyseerd en richtlijnen opgesteld voor toekomstige materialen. Eerst wordt bestaand materiaal geëvalueerd door leerkrachten uit het speciaal onderwijs (Koninklijke Visio en Bartiméus) die eerder gewerkt hebben met verschillende programmeermaterialen. Dit wordt in kaart gebracht door middel van een focusgroep interview (WP1a). Daarnaast wordt bestaand programmeermateriaal door visueel beperkte basisschoolleerlingen getest op toegankelijkheid, gebruikersvriendelijkheid en programmeerkenmerken. Dit wordt gedaan door middel van de constructieve interactie methode, waarbij duo's, terwijl ze interactie hebben met het materiaal, elkaar uitleg geven over wat te doen om een specifieke taak te volbrengen. Met deze methode krijgt de onderzoeker toegang tot het cognitieve proces van de gebruiker(s) (WP1b). Deze twee studies leiden tot de zogeheten SVIC-indicator (SVIC: Suitability for Visually Impaired Children), een score die aangeeft hoe toegankelijk en gebruikersvriendelijk het geteste materiaal is voor visueel beperkte kinderen (WP1c). Op basis van de SVIC-indicator wordt een aantal materialen aangepast om toegankelijkheid en/of gebruikersvriendelijkheid te vergroten (WP2a). Door middel van de peer tutoring approach wordt

daadwerkelijk onderzocht of de aangepaste versie van het programmeermateriaal tot een verbetering heeft geleid ten opzichte van de originele versie of een versie met andere aanpassingen. In deze methode, instrueert het ene kind (tutor) het andere kind (tutee); de tijd die het de tutee kost om de opdracht te volbrengen en de verbale uitingen en gedragskenmerken van frustratie, verveeldheid en plezier zijn een indicatie van de bruikbaarheid van de materialen. De mate van verbetering ten opzichte van de originele versie geeft inzicht in hoe bestaande materialen kunnen worden aangepast om te voldoen aan eisen van toegankelijkheid en gebruikersvriendelijkheid voor visueel beperkte leerlingen. Aan deze studie nemen zowel visueel beperkte leerlingen als ziende leerlingen in het regulier onderwijs deel. De materialen worden hiermee niet alleen getest op bruikbaarheid voor visueel beperkte leerlingen, maar ook in hoeverre het materiaal geschikt is voor ziende leerlingen. Hiermee willen wij voldoen aan de principes van inclusief ontwerp. Wanneer de materialen ook geschikt zijn voor ziende kinderen, is de kans groter dat dat materiaal geschikt is voor CT lessen in zowel speciaal als regulier onderwijs dat bijdraagt aan inclusief onderwijs (WP2b). De inzichten verkregen in bovenstaande studies worden vertaald naar richtlijnen en aanbevelingen, waarbij een lijst wordt opgesteld met geschikte materialen en mogelijke aanpassingen van deze materialen. De verspreiding van de lijst gebeurt onder andere door middel van een interactief platform. Bezoekers van dit platform kunnen niet alleen kennis opdoen over geschikte, inclusieve programmeermaterialen voor visueel beperkte leerlingen, maar kunnen ook hun ervaringen en evaluaties met geteste en nieuwe programmeermaterialen delen (WP2c). Met deze richtlijnen en concrete voorbeelden beogen we Computational Thinking en programmeeronderwijs ook voor visueel beperkte leerlingen in zowel het speciaal als het regulier onderwijs mogelijk te maken.

**Projectleden / Project members**

**Aanvrager / Applicant**
Naam: Dr. ir. F.F.J. Hermans
Organisatie: Technische Universiteit Delft

**Projectleden / Project members**
Naam: Dr. ir. F.F.J. Hermans
Organisatie: Technische Universiteit Delft – Elektrotechniek, Wiskunde en Informatica
Rol: projectleider en penvoerder

Naam: Prof. dr. J.E.J. Schmitz
Organisatie: Technische Universiteit Delft – decaan
Rol: bestuurlijk verantwoordelijke

Naam: Dr. M.O. Hartendorp
Organisatie: Hogeschool Saxion – Toegepaste Psychologie & lectoraat Brain & Technology
Rol: mede-aanvrager

Naam: M. Janssen
Organisatie: Koninklijke Visio – directeur onderwijs
Rol: mede-aanvrager

Naam: Ir. W. Voorn
Organisatie: Koninklijke Visio – Onderwijsinnovatie
Rol: mede-aanvrager

Naam: R. Schenk
Organisatie: Bartiméus
Rol: mede-aanvrager

Naam: E. de Quartel
Organisatie: Bartiméus
Rol: mede-aanvrager

**Probleemstelling / Problem definition - 3696/5000 characters**

Technology is playing an ever-increasing role in society. Therefore, it is important that children are exposed to technology from a young age. It is generally believed that child-technology interactions should not be limited to a consumer role, it is mportant that they understand the way these interactions work. Some of them should even be equipped with the skills to create technology themselves. For them to be prepared for the future they require skills and insights into the functionality of computers. They should be digitally literate. Digital literacy is the combination of digital skills, including Computational Thinking.

As an overarching term for the knowledge and skills that children need to successfully interact with computers, the concept of 'Computational Thinking' was introduced. Being more than just programming, Computational Thinking (CT) is defined as: "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer can carry out" (Wing, 2014).

Several lesson methods, plans and programming environments are already available. These range from commercial robotics sets like LEGO Mindstorms and the BBC Micro:bit to the academically created, free and open source programming language Scratch. In addition, there is a plethora of books, websites and online games and puzzles aimed at teaching children programming skills.

However, both the programming environments and the broader CT teaching techniques rely on visual abilities of children. Programming languages for children are often block-based (Glinert, 1986), meaning that programs are created by arranging blocks. Output also regularly has a visual form, such as digital games or animations. In some cases, robots are involved in the execution stage: for example, to follow a visually represented line or navigate a maze. Moreover, when analyzing whether a program works as desired, the so-called 'debugging', children are often presented with visual clues. Error messages for example, are displayed with popups, or the computer highlights programming blocks that are currently being executed. Some tangible programming blocks have small LED lights to indicate problems in the execution path (Zuckerman et al., 2006). These all use visible information that is necessary to learn the CT basics.

This visual character of most currently existing programming tools for children means that a child with a visual impairment has less or no access to these programming materials. Currently, there are about 2500 visually impaired children at regular schools and about 800 children attending education at Royal Dutch Visio and Bartiméus schools. If programming material is not accessible for them, this will lead to a disadvantage for visually impaired children when learning programming and CT skills. As a result, visually impaired children cannot participate in education. Lack of this skill decreases the chances of this target group on the labor market (see also Goertz et al. (2010) and La Grow (2003) showing that the participation of people with a visual impairment on the labor market is low), partly due to the expectation that the number of jobs in the ICT sector will increase by 17% in the coming years. It is important, especially for visually impaired children, to acquire Computation Thinking and programming skills, to be prepared for the future labor market.

There is no reason to believe that visually impaired children would benefit less from this new form of education. On the contrary, an educational program for visually impaired children can lead to a higher degree of autonomy, as long as the programming curriculum is made accessible for them.

**Relevantie / Relevance – 3914/5000 characters**

There is no complete learning path for Computational Thinking yet, but the national expertise centre for curriculum development (SLO) has made a learning-plan framework. The Dutch House of Representatives agreed in April 2017 to include digital literacy in the curriculum. Digital literacy includes basic computer and information skills, information literacy, information skills and Computational Thinking (Papert, 1980). In the beginning of 2018 teams of teachers and school leaders started with the development of this curriculum. These teams are supported by the Dutch Ministry of Education. They aim at finalizing the new curriculum containing CT skills by the end of 2019. It is important to join the development of the CT curriculum at this moment and not to wait any longer.

Educating children in programming and CT has recently been a topic of interest in many countries. From September 2014, all schools in the United Kingdom now teach programming to children aged five and older. In the Netherlands, already thirty percent of the primary schools indicates they have CT lessons incorporated in their education to some extent (Kennisnet, 2017). As such, it is likely that CT will be taught in almost every Dutch classrooms in the near future.

The majority of children with a visual impairment attend regular education (about 2500 pupils in regular education and about 800 pupils in special education). It is therefore important to focus on the materials used by regular schools. In this project, it is about adjustments or additions to existing programming materials, instead of the development of new materials. In the development of inclusive programming materials, we focus on children with a visual impairment, but also test the materials with sighted children to enhance inclusion.

It is important that all children, including visually impaired children, can benefit from this change in education. This is not just a matter of inclusion but also of independence. The visually impaired community is drastically underrepresented in the field of computer science (e.g., Stefik et al., 2011). While large technology corporations such as Apple Inc. and Google do employ visually impaired coders, these individuals are the exception to the rule rather than a subpopulation. One issue that highlights the importance of the proposed work is the very visual nature of coding, from input, to the spatial (synonymously visual) nature of algorithms or scoping a loop, to output. While attempts have been made to readdress this imbalance in representation, these have mainly been at the mature coder level for individuals who have already developed CT strategies and approaches.

Current vacancies on the labor market request for information technology (IT) knowledge and skills and it is expected that this request will grow the next decades. This means that many jobs are and will be available requiring programming and CT skills. Moreover, unemployment among visually impaired persons is relatively higher in comparison to the total labor population (Goertz et al., 2010). In addition, a higher underemployment rate is noted for visually impaired individuals compared to the total population (La Grow, 2003), meaning that they have a job that requires a lower level of education or skills than the sighted. Expanding the skills of visually impaired children with programming skills and CT will broaden their perspectives on the labor market. If the children could be taught CT and programming skills, they are more likely to find a paid job that matches their level of education.

In addition, this research is in line with the UN Convention on the Rights of Persons with Disabilities (CRPD, 2016). Part of this convention is the aim of inclusive teaching materials that are accessible to both children with and without a visual disability. By offering them the same teaching material, the principle of inclusive education is met.

**Kennisoverdracht, implementatie, bestendiging / Knowledge transfer, Implementation, consolidation – 4799/5000 characters**

We consider different ways of implementing inclusive programming materials to enhance Computational Thinking (CT) skills and knowledge for visually impaired children.

First, the outcomes of the studies and the guidelines for developing inclusive programming materials and examples of inclusive programming materials will be distributed through different platforms, such as LessonUp and EduVIP. EduVIP is the national portal for all educational topics that are important for visually impaired students and their teachers. LessonUp is a platform for free lesson materials for all Dutch schools, considering that most visually impaired children follow regular education

In addition, an interactive platform is created as a result of this project on which stakeholders, such as teachers, parents and children can share experiences and evaluations of the accessibility and usability of programming materials.

Second, teachers who are participating in our research, gain experience with programming materials and CT assignments. These teachers can act as ambassadors of CT and programming skills; they can inform their colleagues about this project, they can explain what CT means and can explain the importance of it. Thus, an indirect, but important side effect of participation in this project means more support among teachers at the schools of Royal Dutch Visio and Bartiméus for CT and programming. In addition, pupils participating in these studies also have the opportunity to experience with programming materials and therefore, with CT. Introducing both children and teachers to programming, makes it more likely that the implementation of the CT curriculum at the Royal Dutch Visio and Bartiméus schools in the near future will succeed.

Third, Novum, the support foundation of Royal Dutch Visio, has funded a project for the development of a curriculum for CT. The Novum project is closely related to this research and results from both projects will reinforce one another. The project of Royal Dutch Visio involves the development of a primary school curriculum, lesson letters and teacher's manuals for CT.
On the one hand, the Novum project can benefit from this research. For a successful implementation, it is important the right programming materials are used. Therefore, the results of the current proposal will help the developers of the new CT curriculum to select appropriate programming materials for the visually impaired children. This means there is direct ground for the results of this project.
On the other hand, the current research will also benefit from the Novum project. Due to the Novum project, teachers already are familiar with CT and programming. This means that those teachers are willing to participate in the current research and are more likely to actually incorporate the outcomes of this research in their classroom.

Fourth, Royal Dutch Visio and Bartiméus have considered inclusive education and in particular CT and programming, as high priority. Therefore, our research touches upon issues that are relevant for these expert centres. It's therefore, very likely that the outcomes of this research will actually be implemented. They have agreed on distributing knowledge from this research, such as spreading knowledge to their schools, supporting teachers in how to implement this knowledge in their lessons. These planned actions indicate strong commitment by Royal Dutch Visio and Bartiméus to actually implement the outcomes of this study in their schools.

Fifth, the results of this research will be published in both scientific and technical journals and will be presented on (inter)national conferences on the topic of vision, education and/or computer science. These publications can lead to schools and other organisations to contact us for further collaboration on this topic and therefore, to the widespread implementation of inclusive CT education. Furthermore, Royal Dutch Visio considers organising an (international) conference on education for visually impaired children by the end of 2019, where the insights from this research can be distributed and be a platform for discussions on inclusive programming education.

Finally, the PI of this proposal is well known in the Netherlands for creating programming materials. For example, 500 children participated in her online programming course for children which was covered in the media extensively[1]. She also developed an online programming course for elementary school teachers. Her network can help us reach children and teachers interested in programming, including visually impaired children.

**Doelstelling / Objective – 2212/2500 characters**

The aim of this proposal is to gain knowledge on what makes programming materials for children inclusive for visually impaired children. Since we aim for inclusive programming materials, we start from existing programming materials. Especially, the materials that are already widely used and have proven to enhance Computational Thinking (e.g. Scratch and Micro:bit) are tested and improved on inclusion for visually impaired children. Our research gains insights in what makes programming materials inclusive for visually impaired children. These insights will be translated into guidelines and recommendations for selecting and developing (future) programming materials for both regular and special education.

Two research questions arise from this objective:

1. What existing primary education programming materials are user-friendly and accessible for visually impaired children, taking the types and severity of their visual impairment into account? [Workpackage 1]
2. Which modifications to existing materials will improve the suitability of existing programming materials for children with different visual impairments? [Workpackage 2]

The target group of our proposal will be visually impaired children (blind and low -vision), ranging from four years old to about twelve years old (covering all grades of primary school). In addition, sighted children will be involved in the process of testing adjusted materials to make sure the final product is created for both visually impaired and sighted children to encourage interaction between visually impaired and sighted children in a classroom following the principles of Inclusive Design. The visually impaired children attend schools of Royal Dutch Visio and Bartiméus (special education) or follow regular education. Sighted children in regular education function as participants in this study to test different programming materials for inclusion. These children are either classmates of a visually impaired child or are in a class that wants to introduce programming lessons in its curriculum. The selection of the latter is based on a match to the visually impaired children participating in this study on level of education (i.e. age and group).

[1] https://www.edx.org/course/scratch-programmeren-voor-kinderen-8-delftx-scratchx-0

**Plan van Aanpak / Strategy – currently about 50.000/60.000 characters**

## Background

*Programming for children*
Programming for children is not a new area. One of the first languages aimed at children was LOGO, created in 1967 (Abelson et al., 1974). LOGO allows children to program a small triangle which can act as a pen, with which they can create drawings, supporting core concepts in programming like variables, lists and recursive functions. More recently, programming education for children has gained more universal and significant interest leading to the development of new programs for children. A few very successful examples are code.org, which has been used by 11 million children, and Scratch whose public repository of programs contains over 15 million programs.

There are also many “unplugged” initiatives, to teach children programming without the use of a computer, like GoldieBlox and RoboTurtle. The biggest breakthrough in putting programming education on the map has been the fact that the United Kingdom has made programming education mandatory for primary school children from the educational year 2014-2015.
All these developments suggest the growing impact of programming education for young children.

*Block-based programming languages*
Learning to program has never been more popular. Code.org for example, a world-wide initiative that encourages kids to learn programming, has been tried by over 100 million kids worldwide. Code.org uses Blockly, a ‘block-based’ language: a programming language that combines a drag-and-drop interface with text fields. A second successful example of a block-based language is Scratch by MIT, whose public project repository currently hosts over 12 million projects.

Block-based programming languages are a special form of *visual programming languages*: languages that let users manipulate program elements graphically rather than by specifying them textually. However, block-based languages also use some successful aspects of text-based languages such as limited text-entry and indentation, and are as such closer to ‘real’, textual programming than other forms of visual programming, like dataflow languages.

Block-based languages have a clear potential to be a great tool for introductory programming education, outperforming text-based languages in some cases (Price and Barnes, 2015). Therefore, the block-based programs seem very suitable for (young) children to acquire programming skills and hence Computational Thinking. Nevertheless, block-based programs require vision and visual overview.

*Tangible Programming for children*
In addition to computer-based (i.e. “plugged”) and unplugged teaching methods, tangible interfaces have been explored and continue to gain momentum. One of the first tangible

approaches was Algoblocks (Tanimoto, 1980) consisting of large blocks that could be connected to a computer and used to create sequences. A more recent approach is Tern (Horn et al., 2009), which provided children with big wooden blocks, resembling young children's playing blocks. These blocks can be connected to one another with wooden pins, after which a computer recognizes the blocks and uses the sequence to control an output robot. Flow blocks is another tangible alternative that aims to teach children causality by chaining electronic blocks on which the control flow is shown (Zuckerman et al., 2006).

These initiatives are a promising means to teach children programming and CT. The creators of Tern noted that programming when not behind a computer had advantages for children's interaction. Tangible interfaces may also have the advantage of drawing more reticent children into programming, especially girls (Horn et al., 2012). This fits existing research indicating that a physical nature invites sharing and exploration. In addition to these academic efforts, recently commercial tangible playing systems were released, like KIBO, Google Bloks, SAM labs and Osmo.

However, existing tangible programming systems aimed at children, despite being tangible, still rely heavily on vision. To create a program, blocks have to be connected with small pegs, and they often come with text and symbols on them. Output, like input is also visual. For example, Tern and KIBO use blocks to control a robot. FlowBlock and SMART labs uses LED lights which are embedded in the blocks. These lights change their color, to help children understand the program's execution. Although a tangible interface appears to be a feasible solution for visually impaired children to be taught programming and CT, the current developments in the direction of tangible interfaces require vision. They require adaptation to be accessible to the visually impaired.

*Sound as output*

There are several programming languages that can be used to create sounds or songs. The above named language Scratch can also produce sounds, in addition to moving 'sprites'. There are also languages aimed solely at makes sounds, of which Sonic Pi is currently the most well-known one. It is used both for introductory programming courses and professional music making.

An auditory solution allowing visually impaired children interact with games is using 3D sound in a virtual reality environment. Sanchez and Aguayo (2005) have tested different audio-based virtual environments and have shown that these environments led to visually impaired children performing complex tasks involving abstraction, such as spatial orientation and laterality. A disadvantage of 3D sound is that it limits interaction with others, since the task should be conducted wearing headphones. This disadvantage is also recalled in other studies concerning interfaces requiring headphones (e.g. Bradley and Dunlop, 2002). Wearing headphones will block environmental sounds out. Nevertheless, in a game setting, the visually impaired children were very exciting about the use of 3D sound (Sanchez and Saenz, 2007). Sound as output seems very promising for designing inclusive programming materials.

*Programming materials for visually impaired children*

Recent initiatives have arised in developing suitable programming materials for visually impaired children (e.g. Story Blocks (Koushik and Kane, 2017), Blocks4all (Milne et al., 2017), and Donnie Robot (Marques et al., 2017)). The materials were based on designs shown to be effective for children, such as a block-based structure and robots. These materials tackle several issues revealed by visually impaired programmers previously (Albusays and Ludi, 2016; Dini et al., 2006). Examples of issues are the drag-and-drop interface of most block-based programming environments, the lack of overview due to the use of a screen reader or magnification, and the visual way of providing output. Text-to-speech, sound output and vibrations as feedback were introduced as solutions to the issues experienced by blind users of general programming languages. However, most of these recent initiatives are not available yet and more importantly, they do not cover the scope of user age, learning objectives and work methods necessary for functioning properly in education.

*Inclusion*

Increasing awareness is emerging in society on inclusion and participation. In 2016, the Dutch government agreed on following the UN Convention on the Rights of Persons with Disabilities to endeavor inclusion, personal autonomy and full participation. In 2012, a couple of large organizations have founded the Web Accessibility Initiative (by the World Wide Web Consortium, www.w3.org) to enhance accessibility for all on the web. A number of general guidelines have been presented following principles of Inclusive Design (e.g. for a definition of Inclusive Design, see BS 7000-6:2005). In general, these guidelines can be divided into guidelines for a perceivable, understandable, operable and robust web environment for people with different types of disabilities. These principles are applicable to web sites and web applications, but also to other types of software applications.

The W3C encourages to consider evaluation by the end users at several stages of the development process. This prevents developers to have to adapt the web environment afterwards, saving many costs. Nevertheless, many web applications and software tools in general, are not designed with people with disabilities in mind. This means that still many digital tools are being tested for accessibility when they are already commercially available (e.g. Caruso et al., 2017).

Most studies on accessibility for visually impaired persons involve predominantly blind users. However, someone who is blind or has low-vision addresses different issues and is in need for different requirements to make a design accessible for them. In addition, within the group of people with low vision, the variety is large on their access needs (e.g. Dini et al., 2006). In addition, there are more tan three times as much people with low vision than those suffering from blindness across the world (WHO, 2017), making it even more relevant to consider accessibility and usability testing for both the blind and people with low vision.

In conclusion, several initiatives have been introduced to teach children CT and programming skills. These initiatives are more or less accessible for visually impaired children. Some of them are even created with the blind child in mind. Due to gaining popularity of CT and programming in primary education, it is important to act now and guide teachers in selecting suitable materials for their pupils. And if not suitable yet, guidance is provided in how to adjust materials to make them more suitable for visually impaired children taking their type and severity of visual impairment into account.

**Research question(s)**
The aim of this research project is to understand to what extent is existing programming material for children inclusive for visually impaired children?

Therefore, two research questions have been defined:

1. On which aspects of accessibility, usability and programming features is existing programming material for primary schools suitable for visually impaired children, taking type and severity of the visual impairment into account?
2. How can existing programming materials become more inclusive for visually impaired children?

The first research question will provide insights into which material is suitable for visually impaired children and, above all, what makes the material suitable. Suitability is understood in the broadest sense of the word. The extent to which the material is suitable for a variety of types and severity of visual impairments shall be considered. The aim is to find suitable material that contributes to inclusive education.

The results will be translated into a so-called SVIC-indicator (Suitability for Visually Impaired Children), a matrix summarising the level of suitability on aspects of accessibility, usability and programming characteristics. Such an indicator helps in selecting proper material. Moreover, when considering new, not tested, material, the SVIC-indicator can be used to explore how inclusive the material could be.

The second research question focuses on giving insights into how material can be adjusted leading to more inclusive material. The SVIC-indicator is used as a starting point for modifying existing programming materials. Adjusted programming materials will be tested in comparison to the original programming material, including both visually impaired and sighted children as participants. The latter is important to draw conclusions on the level of inclusion of the different materials. Eventually, these insights are translated into guidelines. In this way, these guidelines are also useful for material yet to be developed. As such developers and producers can use them to make future material suitable for all pupils.

*Existing programming materials*
Many programming materials for children are already available and widely used in many classrooms worldwide including the Netherlands. Teachers have a wide variety of materials to choose from that are (freely) available for them to use in CT lessons. In the current proposal, we,

therefore, endeavour on delivering an overview of suitable programming materials for visually impaired children based on existing programming materials. Some materials will be adjusted in the current study, but predominantly to gain insights in what modifications could make materials more suitable for visually impaired children to interact with the programming material. These insights are translated into guidelines, to enhance more inclusive programming materials in the (near) future. In addition, a great advantage of using existing programming materials, and in particular the widely-used materials, is that they are more likely to be well maintained and affordable (or even freely accessible). At last, the selection of programming materials will not only be based on its common use in education, but also on its variety. The more variety among the different materials tested, the more insights in what is suitable for visually impaired children. These insights that can be used for existing and future materials.

*Degree of inclusion*

The programming materials are tested for inclusion. In the context of this research, this means that the material must be suitable for visually impaired and sighted children.

Inclusion is subdivided into usability (i.e. user-friendliness) and accessibility, with the idea that when the material is user-friendly and accessible to the target group(s) concerned, it is also considered to be inclusive (Queiros et al., 2015; Wegge and Zimmermann, 2007). To determine the extent to which programming material is inclusive, the material is tested in a learning situation in both special and regular education (with one or more visually limited pupil(s) in the classroom).

*Usability.* Usability means the extent to which a product is used by specified users in a given context of use to achieve a specific purpose in an efficient, effective and satisfactory manner.

*Accessibility.* Accessibility means the extent to which a product can be used by as many people as possible. A product is accessible when information for the use of the product is perceivable, if the product can be operated and if the information and operation is understandable (i.e. perceivable, operable, understandable). This concerns the accessibility of the input, feedback and output of the material. The nature of the visual impairment will also be taken into account; for instance, some visually impaired children read braille, others not.

*Programming features.* In addition, educational programming tools not only require accessibility and usability, but it is also important they fulfil specified learning objectives. When considering suitable materials for CT lessons, it is important the material is not only accessible for visually impaired children, but also ensures children are educated in programming and CT skills. Therefore, aside from accessibility and usability, the materials are also tested for their programming features. This supports a teacher to select suitable material for a programming lesson not only on inclusion, but also on CT objective (i.e. content) and lesson method. The programming features follow the five criteria as defined in the TU Delft teaching module[2] for teachers: work method, requirements, programming concepts, prior knowledge and learning objectives.

*Participants*

We ensure ethical procedures will be followed in advance of all phases in the project. This will be done by the ethical committee of the Delft University of Technology and no research will be

[2] https://www.edx.org/course/programmeren-voor-leerkrachten-met-delftx-scratchtx

undertaken before the approval of the ethical committee is obtained. Moreover, the parent(s) or caregiver(s) of the children that will be selected for this research project will be asked for their permission by signing an informed consent. In accordance with the ethical guidelines laid out in the Declaration of Helsinki, this informed consent will state that we will deal in a confidential and anonymous manner with the data collected and both the children and/or the parents will be given the opportunity to stop participation in this project at any time without needing to provide a reason for their decision. The studies are conducted in the classroom during regular lesson time. If children are not permitted to participate in the study, they can still take part in the lesson, but they will not be (video or audio) recorded and their responses will not be taken into account in processing the data. Only data is processed that comes from children who(se) parents or caregivers gave permission to participate in the study.

*Workpackages*

*Workpackage 1 (WP1).* Because of its wide variety, it could be difficult for teachers of visual impaired children to select suitable material, since most teachers are not experts in programming either. The current research aims at providing an overview of existing programming materials and the level of suitability for visually impaired children, taking level of education and type and severity of visual impairment into account. This overview, the so-called SVIC-indicator (where SVIC stands for Suitability for Visually Impaired Children), supports teachers in selecting suitable materials. The SVIC-indicator is based on research conducted in the classroom increasing ecological validity of the testing of existing programming materials.

*Workpackage 2 (WP2).* Subsequently, the SVIC-indicator will be the starting point for adjusting materials that could lead to improvements of the materials on inclusion of visually impaired children. This study will lead to more insights on what aspects programming materials should consist of to be (more) suitable for visually impaired. These insights will lead to guidelines for developing new, more inclusive programming materials.

An overview of the workpackages is provided in Table 1 and 2.
Notably, every subworkpackage ends by a separate implementation section. By doing so, it is emphasized how the outcomes of each (sub)workpackage can actually reinforce teaching Computational Thinking to visually impaired children.

*Table 1.* Overview of workpackage 1

| **WP1. Gaining insights in level of inclusion of existing programming materials for visually impaired children** | **Research method (RM) / Deliverable (D)** | **Participants / partners** |
|---|---|---|
| WP1a. Teachers' experience with programming materials for children | RM: Focus group interview | Teachers working at Royal Dutch Visio and Bartiméus primary schools |
| WP1b. Visually impaired children's interaction with | RM: Constructive interaction | Visually impaired children of Royal Dutch Visio and Bartiméus |

| | | |
|---|---|---|
| programming materials | | primary schools |
| WP1c. SVIC-indicator | D: SVIC-indicator (Suitability for Visually Impaired Children) | Consortium research proposal (TU Delft, Saxion, Visio and Bartiméus) |

*Table 2*. Overview of workpackage 2

| **WP2 – Improving existing programming materials on level of inclusion** | **Research method (RM) / Deliverable (D)** | **Participants / partners** |
|---|---|---|
| WP2a. Adjusting programming materials | D: different versions of programming materials; one original version and one or two adjusted versions | TU Delft and Saxion |
| WP2b. Systematically testing (adjusted) programming materials for inclusion | RM: Peer tutoring approach | Visually impaired children of Royal Dutch Visio and Bartiméus primary schools, and visually impaired and sighted children of regular primary schools |
| WP2c. Defining guidelines | D: Guidelines for inclusive programming materials for children | Consortium research proposal (TU Delft, Saxion, Visio and Bartiméus) |

**WP1 – Gaining insights in level of inclusion of existing programming materials for visually impaired children**

*WP1a. Teachers' experience with programming materials for children*
*Period.* 0-4 months
*Participants*: Seven teachers who work as a teacher at primary schools of Royal Dutch Visio (locations Grave, Huizen and Rotterdam) and Bartiméus will be interviewed on their experiences with existing programming materials. They are teachers who are interested and enthusiastic for introducing Computational Thinking in their lessons but have no background in IT or computer science. The teachers differ in the age of the children they teach, varying from first grade to eight grade. However, they are all teachers in special education. The experience the teachers have with programming materials varies from experimenting with materials introduced in the context of the Novum project (Royal Dutch Visio) to experience with actually adjusting existing materials to make the material more suitable for visually impaired children (Bartiméus). However, the experimenting with programming materials was not done in a systematic way as we aim for in the current study.

*Method.*  A focus group interview will be performed with the teachers at Royal Dutch Visio and Bartiméus schools who have introduced and evaluated different existing programming materials in their classes previously. The materials introduced are mainly existing programming materials for children; some programming materials, however, have already been adjusted to suit the possibilities of the visually impaired children (e.g. tangible adjustments to the tactile reader for the BlueBot). The focus group interview is a good method to gain insights in a target group's experiences, opinion and ideas for future design. Due to the group setting, participants do not only elaborate on their own opinion, but also get inspired by the opinions of others leading to more ideas than in an individual interview.

*Materials.* To conduct the focus group interview, a video camera is used to record the group interview. The benefits and barriers experienced by the teachers when introducing programming materials in their classes previously are discussed for each programming material individually. To enhance discussion, all tested materials are present. This allows participants to memorize each material again, but also to show others what benefits or barriers they have experienced. The latter could reinforce more discussion and exchange of experiences between teachers.

*Procedure.* First, permission is required by the participants to use the video recordings for analysis and further use and possible publication of the data. The data are processed anonymously and confidentially and will only be used for this workpackage (WP1). Next, the participants take place in a circle, with in the middle different programming materials that some of them have tried out in their classes previously. All programming materials are discussed, one by one.

*Analysis.* The video recordings are transcribed and will be analyzed using qualitative data analysis (open, axial and selective coding).

*Implementation.* One strategy for successful implementation is the introduction of peer groups (intervision groups).  As part of this project, Royal Dutch Visio will introduce peer groups for its teachers who have joined the focus group interview, starting during the two year period of this study. Once every month, teachers come together and share their experience with (new) programming materials, are being introduced to new developments of programming materials/lessons, and support each other. The first peer group meeting will be initiated by the consortium of this proposal, in which the results of the first study will be presented to the participants. These teachers can be seen as innovators and/or early adopters (Rogers, 1995) and it is therefore important to give them a key role in implementing knowledge on suitable programming materials for visually impaired children.

The peer group meetings that will take place once every month will take two hours and will be supervised by the coordinator of Royal Dutch Visio. The first hour concerns evaluating, reflecting, sharing. The second hour concerns introducing new materials, experiencing and testing. Participants of these meetings will be motivated to continue with the programming lessons (first hour) and will be inspired to connect to new developments (second hour).

Due to the active participation of this group of teachers, they will share their enthusiasm and experience with other teachers, who are more concerned to be part of the early and late majority who adopts innovation (Rogers, 1995). It is expected, more and more teachers will consider the use of technology, CT and programming knowledge and skills in their classes (also when they teach in other disciplines such as mathematics and Dutch language). Eventually, these peer group meetings will lead to a widely-spread acceptance, adoption and implementation of CT at the schools of Royal Dutch Visio and Bartiméus (REFS).

*WP1b. Visually impaired children's interaction with programming materials*
*Period*: 4-10 months
*Participants*: About forty-five visually impaired children of Royal Dutch Visio and Bartiméus schools participate in testing several programming materials for accessibility and usability. These children are in the groups of the teachers interviewed in WP1a. In every group there is about six to seven pupils. Most of these classes have already participated in pilot studies regarding Computational Thinking. Teachers will also be involved by providing time during their lessons for us to test the materials. Nevertheless, these teachers are already involved in CT lessons, and will make time available in their lessons. Moreover, the testing can be considered as a lesson in CT, in which all children in each class will be involved. This means that participants do not have to spend extra time outside regular lessons to participate in this study. The same accounts for the teachers involved. Children and their parents / caregivers will be asked to give permission for participation in this study. This will be asked by informing parents, teachers and children about the goal and procedure of this study. If some children and their parents do not want to participate in this study, they can still join these lessons. Their contributions in the class, however, will not be used for data analysis.
*Method.* In this study, the constructive interaction method is used. In constructive interaction, two participants are asked to perform a task together interacting with one of the programming materials tested. The interaction between the two participants in solving the task provides insights in the mental models of the participants and usability experience by the participants. In usability testing with adults, the think aloud method is more commonly used. However, for (young) children it is assumed that constructive interaction is more effective than thinking aloud due to the more natural setting of the setup in constructive interaction in comparison to the less natural situation of verbalizing ones thoughts and decisions in the thinking aloud protocol. When collaborating with peers in performing a task, children automatically think aloud and let its peer (and therefore the researcher) a preview of their thoughts, mental concepts and understanding of the program they work with. In addition, the younger the children, the harder it is for them to verbally conceptualize their thoughts. Interacting with peers in performing tasks is a natural setting at primary schools and most programming materials for children naturally reveal collaboration and interaction between children. Thus, it is expected to lead to more natural behavior enhancing ecological validity of the test situation.
*Variables.* Between-subject variables are taken into account: eye condition and severity of eye condition, age and group are taken into account. Furthermore, the amount of experience with the programming material(s) tested is taken into account (expressed in number of hours), because familiarity with a program could have influence on the level of usability and accessibility problems experienced.
*Materials:* About a dozen different programming materials will be selected for investigation. Selection of programming materials is based on the following three selection criteria: 1. Variability - a wide range of types of programming materials, mainly varying in visual, audible and tangible feature is selected. They represent a broad range of possibilities for the children, leading to more

insights in what possibilities there are to make programming materials accessible for visually impaired children. However, materials can also vary on plugged versus unplugged programming and other aspects, such as difficulty and extension of the material; 2. Availability, such as costs and delivery time (notably, all partners of this consortium own already different programming materials that can be used for this study; however, costs are budgeted to purchase some programming materials for this study); and 3. Commonly-used materials at regular schools. This is important to increase the implementation of the current material. Widely-used materials are more well-known and are more likely to be developed further and maintained than materials that are not often used.

*Procedure.* The interaction by pairs of visually impaired children with the selected programming materials will be studied in regular lessons. Therefore, all children in a class participate in the study. However, their data will only be used when given permission. Every class participates in three lessons. In every lesson of an hour, two different programming materials are tested on accessibility, usability and their programming features. Children will all perform one task. Students of Saxion University of Applied Psychology will assist the postdoctoral researcher in this procedure of collecting data, performing observations and transcribing qualitative data as preparation for the qualitative data analysis. As preparation, the student assistants receive a training in conducting observations in a constructive interaction design and in transcribing qualitative data from video.

*Analysis.* The video recordings are transcribed and will be analyzed using qualitative data analysis (open, axial and selective coding). The data are analyzed following the sensitized concepts of accessibility (i.e. perceivable, operable and understandable) and usability (i.e. effectiveness, efficiency and satisfaction).

*Implementation.* By participating in this study, the visually impaired children are exposed to programming and CT. Moreover, due to the study design, the children will be actively involved in experiencing with programming skills and materials. In general, children respond with great enthusiasm to CT lessons. This is also our experience when testing programming materials in previous pilot studies at Royal Dutch Visio and Bartiméus schools. The programming materials are engaging, challenging and can even lead to more self-esteem since it gives children a feeling of competence. Due to this enthusiasm, some children want to continue programming and start programming at home. In addition, due to their enthusiasm, parents and teachers can no longer stay behind and have to introduce programming and CT in their (school/home) environment too.

*WP1c. SVIC-indicator*

*Period*: 10-12 months

*SVIC-indicator:* The results of WP1a and WP1b are combined and converged into the so-called SVIC-indicator. This matrix contains aspects of accessibility, usability and programming features on the x-axis and (parts of) the tested programming materials on the y-axis. The aspects on the x-axis are based on the outcomes of the focus group interviews, definitions of accessibility and usability, and on the list of the five programming features developed by Delft University of Technology. In the cells of the SVIC-indicator a rating is assigned that indicates the scoring of a programming material on that specific aspect. The scoring scale contains three points, namely good (2), sufficient (1) and insufficient (0). The individual scores will result in a total score that

leads to a ranking of suitable programming materials on accessibility, usability and programming features for visually impaired children.
*Implementation*: the SVIC-indicator is published on open source libraries, such as LessonUp (FutureNL), EduVIP and shared in (online) newsletters of Royal Dutch Visio (the Kennisportaal) and Bartiméus. In addition, the SVIC-indicator is presented in a peer group meeting (see 'implementation WP1a') being attended by CT teachers of Royal Dutch Visio schools. This helps these teachers to select suitable programming materials for their future CT lessons. In addition, if they consider programming materials not being tested in this study, they can use the SVIC-indicator matrix to rate that material, leading to the use of more suitable programming materials for visually impaired children in CT lessons.

*Deliverables WP1 - Gaining insights in level of inclusion of existing programming materials for visually impaired children*
*Period*: After one year

1. SVIC-indicator published on LessonUp and EduVIP and presented through internal (online) newsletters of Royal Dutch Visio and Bartiméus
2. Scientific publication and an abstract/presentation at an (inter)national conference
3. Publication in a technical journal on education and/or technology

**WP2 – Improving existing programming materials on level of inclusion**
*WP2a. Adjusting programming materials*
*Period*: 12-16 months
*Materials*: Based on the SVIC-indicator, three programming materials will be selected that have high potential to be improved on inclusion for visually impaired children. Changes could be made to the input interface (e.g. adding labels to icons on the screen to make the screen accessible for screen readers), the feedback system (e.g. instead of visual popups, auditory cues are used), or the output (e.g. creating music instead of making an avatar move on the screen). Only one feature of the programming material is changed at a time, to ensure that improvements on level of inclusion can be reduced to that particular change. In total, one to three adjusted version of the original material arise from this modification phase.
Adjusting materials will be done by the postdoctoral researcher who has preferably a background in Industrial Design and/or Software Engineering and will be done in close collaboration with the Faculty of Software Engineering if changes are required to the software of the materials and Applied Psychology to meet the target group requirements.

*WP2b. Systematically testing (adjusted) programming materials for inclusion*
*Period:* 16-22 months
*Participants*: About forty-five visually impaired children in special education, about five visually impaired children in regular education and their sighted classmates and some groups in regular education with no visually impaired classmate are included in this study. The visually impaired children in special education are pupils in the groups of the teachers involved in WP1. Hence, these are pupils from Royal Dutch Visio and Bartiméus schools. The visually impaired pupils in regular education will be recruited via support teachers (in Dutch: Ambulant Onderwijs

Begeleiders). A number of schools with visually impaired children on their school have already expressed their intentions in willing to collaborate on making programming lessons inclusive (these are schools of Openbaar Onderwijs Groningen, the Netherlands). In addition, the PI is currently testing different programming materials on several primary schools in regular education. Some of these schools will also be asked to participate in this study.

*Method:* To what extent the adjustments to the programming materials leads to improvements on inclusion is studied following the peer tutoring approach (Höysniemi et al., 2003). In this approach, one child is instructed on how to perform a particular task by an (adult) instructor. Next, the child becomes a tutor in using the software program. As a tutor, the child will explain how to use and interact with the software program to another child who is referred to as the tutee. How easy the tutee learns to interact with the programming material by the instructions of the tutor indicates the level of usability and accessibility. Though the peer tutoring approach is not the most common method used to investigate usability, it seems a promising method for this study. First, in a pilot study conducted in January 2017 by the consortium with visually impaired children, children were asked to interact freely with different programming materials. Automatically, children were helping each other by instructing the others how to use a specific program they have used before. Naturally, children tend to instruct another, and the latter also accepts and enjoys being instructed by the other. Second, this method makes it possible to test different programming materials at once, saving a lot of spare lesson time of the schools. This is possible because several children can start as a tutor for different programming materials and can continue testing other programming materials as tutees (see also WP2b - Procedure). Third, the method is investigated for children in the age from five to nine years old, showing to be a suitable method for usability testing with (young) children.

In previous research using the peer tutoring approach, bad design according to usability criteria was reflected in the child's behaviour. This was visible by the tutee being distracted, did not want to continue further and/or said the assignment was too difficult. Other indicators of usability were the ability to fulfil the assignment (efficacy) and the time needed to finish the assignment (efficiency). In the current study, the verbal expressions and behaviour of the tutees are analysed on signals of joy, frustration and boredom (satisfaction). Furthermore, finishing the assignment is recorded (efficacy). And last, the time a tutee needs to finish the assignment is recorded (efficiency).

*Variables*: The independent variables are: Version (original or adjusted version), Type of Education (special or regular), Inclusion (visually impaired or sighted) and Type of Eye Condition.

*Materials.* For recording the tutees by video and the tutors by audio, a videorecorder is used. In addition, the selection of programming materials tested is based on the results of WP2a. About three programming materials are selected and adjusted. This means that there will be three original versions of programming materials and a small number of adjusted materials.

*Procedure:* First, all children in a class are explained one specific task. For example, program a square. Next, for each (version of) programming material one child is taught by the instructor (researcher and student-assistants) how to perform that specific task using that specific programming material (either original or adjusted). When taught and practiced, the children have become so-called tutors. Subsequently, the tutors teach a tutee how to perform the task using that specific material. The tutees are the tutors' classmates. The tutor can only explain how to interact

with the material and is not allowed to show it by interacting with the material him/herself. The instructor records the time in seconds the tutee needs to perform the task. If the tutee has performed the task, the tutee becomes the tutor and a new classmate becomes the tutee. The former tutor becomes tutee for one of the other programming materials. While performing the task being instructed by the tutor, a tutee is recorded by video, and the tutor is recorded by audio. The task will be the same for the different versions of the same programming material. A learning effect is expected when performing the same task for the different versions of that programming material. Nevertheless, because every tutee performs the tasks in a different order, the learning effect is automatically counterbalanced.
*Analysis:* Several data are analyzed in this study. First, the number of assignments finished are compared for the different (adjusted versions of) programming materials using General Linear Model Repeated Measures including the independent variables Version and Inclusion. Furthermore, the time spent to run the assignment measured in seconds; the more accessible (adjusted) programming material is, the shorter the recorded time, again analyzed using the General Linear Model Repeated Measures. Eventually, the video and audio recordings are analyzed using qualitative data analysis, particularly focusing on verbal expressions and behavior of enjoyability, frustration and boredom.
*Implementation.* By participating in this study, teachers in regular education will experience that programming lessons are possible using the same materials for sighted and visually impaired child(ren). Moreover, they will have the experience that this material is suitable for encouraging collaboration between sighted and visually impaired children. Both experiences are likely to reinforce a teacher to continue on programming lessons for all children in its class. In addition, teachers in regular education are pointed out to the SVIC-indicator, making them aware of what programming materials are suitable for visually impaired children and on what aspects of accessibility, usability and programming features a teacher can pay attention when selecting other programming materials than tested in this study.

*WP2c. Guidelines for inclusive programming materials for visually impaired children*
*Period:* 22-24 months
*Guidelines:* Based on the results of WP1 and WP2, guidelines will be defined to increase the creation of inclusive programming materials for children. A lot of material is currently available and there will be a lot more on the market in the (near) future. In order to be able to use the research results in the future, the results are translated into guidelines and recommendations for (modifications of) CT teaching materials. Guidelines for usability and accessibility already exist for software design/web applications (W3C). However, these guidelines are formulated in a general manner, applicable to a large group of disabilities. In contrast, our guidelines focus specifically on programming materials for visually impaired children, though emphasizing its use in regular education. Notably, the guidelines as a result of this study can be seen as complementary to existing guidelines. We aim for concrete examples on how different aspects of programming materials can be improved on becoming more accessible and/or usable for visually impaired children. By following the guidelines, it increases the chance of deploying suitable material for visually impaired students in an inclusive learning environment. The guidelines and

recommendations can be used by teachers to gain insight into the suitability of (future) materials. They can also be used by developers of new education materials.

*Implementation*: First, the guidelines and (evidence-based) examples will be presented on an interactive platform. Teachers, parents and even pupils will have access to this platform. Here, they can gain knowledge on suitable programming materials for visually impaired children, but can also share their experiences and evaluations of already tested or newly developed programming materials. In this way, selecting suitable materials for programming lessons is both evidence-based and practice-based.

Furthermore, at the end of the project period a meeting is organized. The consortium researchers and employees of Royal Dutch Visio and Bartiméus will attend the meeting, but also teachers of both special and regular education who participated and who are interested in inclusive CT lessons are invited. We also will invite developers and policy makers of CT and programming education, such as SLO, PO-Raad (Primair Onderwijs Raad; elementary education council) and Ministry of Education. By organizing this meeting, we aim for dissemination of the knowledge of and experience with good examples of inclusive programming materials. Attenders of this meeting will exchange, discuss and experience with different (adjusted) programming materials and will be informed about the SVIC-indicator and the guidelines. In addition, a number of children who participated in one of the studies will be invited to demonstrate how they interact with tested and adjusted programming materials. In this way, attenders of the meeting can see how visually impaired children get access to programming. At last, follow-up steps will be discussed to ensure inclusive programming materials are incorporated in CT lessons at primary schools, both in regular and special education.

*Deliverables - WP2. Improving existing programming materials on level of inclusion*

1. Guidelines for developing new inclusive programming materials for children published on an interactive platform
2. List of suitable programming materials with (relatively small) modifications that can be used in regular and special education in CT lessons. The list is published on LessonUp and EduVIP
3. Scientific publication and presentation at an (inter)national conference on vision, education and/or computer science.
4. Publication in a technical journal on the topic of education and/or technology

## Expertise, voorgaande activiteiten en producten / Expertise, prior activities and products – currently 2937/7500 characters

*Expertise*

We believe the scientific consortium has the perfect constitution to execute this proposal.

The Software Engineering Research group has over a decade of experience in end-user programming: programming for non-programmers, and is one of the world's leading groups in this area. Hermans has worked on programming and debugging for sighted children extensively (Hermans and Aivaloglou, 2016, Aivaloglou and Hermans, 2016, Hermans, Stolee and Hoepelman, 2016) and has designed of a popular Dutch teaching Scratch method[8].

Applied Psychology at Saxion University of Applied Sciences educates students on coaching, training, supervising and interviewing. In addition, they are trained in conducting applied research within the field of Psychology, including usability testing, effect studies and evaluation studies. The research group Brain & Technology conducts applied research on the topic of societal participation for people with different disabilities. This concerns topics on increasing autonomy, accessibility and empowerment. Hartendorp works as a lecturer within the research group Brain & Technology, and conducts studies concerning (multisensory) perception and user strategies. Many of these studies are in close collaboration with Royal Dutch Visio.

Bartiméus and Royal Dutch Visio have great experience in teaching to visually impaired children, both blind and low vision. In addition, many teachers are trained in selecting and creating materials suitable for their pupils.

Both Bartiméus and Royal Dutch Visio started with Computational Thinking and programming education. Some specialised ICT teachers of Bartiméus programmed the Micro:bit with python, together with some blind students.

*Prior products and activities*
At Royal Dutch Visio, a group of teachers started with the Novum project at the beginning of 2017. In close collaboration, teachers in special education of Royal Dutch Visio, Delft University of Technology and Saxion University of Applied Sciences have developed several CT lessons for groups 1 to 4 (from four to eight years old) using mainly unplugged programming materials. The teachers tested and evaluated the developed curriculum in their classes. Next to the curriculum activities, a lot of programming materials (e.g. Beebots, Cubetto and Scratch) were tested in a pilot study at Royal Dutch Visio schools. The results of this pilot study were presented at the international conference Vision 2017 in The Hague. Royal Dutch Visio also joined the 'Expeditie Micro:bit' from FutureNL, together with 500 regular school. This 'expedition' was a big challenge for particularly blind students; as well as the mini-computer as the software are not accessible. With some extra hands and additional tools, these kids could also experience the possibilities of the Micro:bit. These experiences with the Micro:bit and other materials were published in the national education magazines COS and Vives.

**Publicaties / Publications (maximaal 7500 karakters)**

E. Aivaloglou and F. Hermans (2016) How Kids Code and How We Know: An Exploratory Study on the Scratch Repository. In Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16). ACM, New York, NY, USA, 53-61.
F. Hermans and E. Aivaloglou (2016) Do code smells hamper novice programming: A controlled experiment on Scratch Programs –In proceedings of the International Conference on Program Comprehension, 1-10.
F. Hermans, K. Stolee and D. Hoepelman (2016) Smells in Block-Based Programming Languages. In Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing.

**Referenties / References (maximaal 7500 karakters)**


H. Abelson, N. Goodman, L. Rudolph (1974). "LOGO Manual". Artificial Intelligence Lab, Massachusetts Institute of Technology.

K. Albusays and S. Ludi (2016). Eliciting programming challenges faced by developers with visual impairments: exploratory study. In Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering, 82-85.

N. Bradley, and M. Dunlop. (2002). Investigating context-aware clues to assist navigation for visually impaired people. In Proceedings of Workshop on Building Bridges: Interdisciplinary Context-Sensitive Computing, University of Glasgow.

G. Caruso, S. Dini, L. Ferlino, S. Panesi (2017). Web 2.0: A new opportunity to create shared knowledge to support visually-impaired children in education and rehabilitation. ICERI2017 Proceedings, 4369-4379.

S. Dini, L. Ferlino, A. Gettani, C. Martinoli and M. Ott (2006). Educational software and low vision students: evaluating accessibility factors. In Universal Access in the Information Society, 6, 15-29.

M. Horn, R. Crouser and M. Bers (2012). Tangible interaction and learning: the case for a hybrid approach. Personal and Ubiquitous Computing 16 , 4, 379–389.

M. Horn, E.T. Solovey, R. J. Crouser and R.J.K. Jacob (2009). Comparing the use of tangible and graphical programming languages for informal science education. In Chi ’09 32 , 975.

E. Glinert (1986). Towards "Second Generation" Interactive, Graphical Programming Environments, Proceedings of the IEEE Workshop on Visual Languages.

Y. Goertz, B. van Lierop, I. Houkes and F. Nijhuis. (2010). Factors related to the employment of visually impaired persons: A systematic literature review. Journal of Visual Impairment and Blindness, 104(7), 404.

J. Höysniemi, P. Hämäläinen and L. Turkki (2003). Using peer tutoring in evaluating the usability of a physically interactive computer game with children. In Interacting with computers, 15(2), 203-225.

V. Koushik and S.K. Kane (2017, October). Tangibles+ Programming+ Audio Stories= Fun, In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, 341-342.

S. J. La Grow. (2003). Employment among working-age members of the Royal New Zealand Foundation for the Blind. Journal of Visual Impairment and Blindness, 97, 429-433.

G.H. Marques, D.C. Einloft, A.C. Bergamin, J.A. Marek, R.G. Maidana, M.B. Campos and A.M. Amory (2017). Donnie robot: Towards an accessible and educational robot for visually impaired people. In Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 1-6.

L.R. Milne, C.M. Baker and R.E. Ladner (2017). Blocks4All Demonstration: a Blocks-Based Programming Environment for Blind Children. In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, 313-314.

S. Papert (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

T. W. Price and T. Barnes (2015). Comparing Textual and Block Interfaces in a Novice. In Proceedings of the ACM International Computing Education Research (ICER).

A. Queirós, A. Silva, J. Alvarelhão, N.P. Rocha, A. Teixeira (2015). Usability, accessibility and ambient-assisted living: a systematic literature review. In Universal Access in the Information Society, 14(1), 57-66.

E.M. Rogers (1995). Diffusion of innovations. 4. New York: Free Press.

J. Sanchez and F. Aguayo (2005). Blind learners programming through audio. In Proceedings of CHI '05, 1769–1772.

J. Sánchez and M. Sáenz (2007). Usability of Audio-Based Virtual Environments for Users with Visual Disabilities. In Virtual Reality and Human Behavior Symposium, LAVAL Virtual, 18-22.

A. Stefik, C. Hundhausen and D. Smith (2011). On the Design of an Educational Infrastructure for the Blind and Visually Impaired in Computer Science. In Proceedings of the 42nd ACM Technical symposium on Computer science education, 571-576.

S. Tanimoto (1990). VIVA: A visual language for image processing. Journal of Visual Languages Computing, 127–139.

K.P. Wegge and D. Zimmermann (2007). Accessibility, usability, safety, ergonomics: concepts, models, and differences. In International Conference on Universal Access in Human-Computer Interaction, 294-301.

J. Wing (2014). Computational Thinking Benefits Society. In 40th Anniversary Blog of Social Issues in Computing.

O. Zuckerman, T. Grotzer and K. Leahy (2006) Flow blocks as a conceptual bridge between understanding the structure and behavior of a complex causal system. In Proceedings of the 7th international conference on Learning sciences (ICLS '06). International Society of the Learning Sciences, 880-886.

**Bijzondere gegevens (maximaal 2325/7500 karakters)**

Though we involve both visually impaired children and teachers of visually impaired children to participate in the studies this will not be experienced as such, since the two studies involving visually impaired children occur during regular lesson time. Moreover, the experimental setup has been chosen in such a way that it looks like a real lesson due to the assignment character of the studies. Previous experience in pilot studies have shown that children actually enjoy these lessons and that teachers experience this not as an extra burden for them. Notably, pupils that do not want to participate or partners/caregivers of children that do not want them to participate, are excluded from data analysis. These pupils can still participate in the classroom as usually, but will not be observed or recorded. This makes it a very ecological and feasible approach.

Both Bartiméus and Royal Dutch Visio have started with Computational Thinking and programming lessons. All the involved teacher are regular teachers, without any knowledge of computer science or programming languages. At this moment these lessons aren't part of the regular curriculum, both expert centres however attach great value to these teaching topics, so they have been exploring adding them to the curriculum recently. Royal Dutch Visio participanted in three free programming lessons from the 'Expedition Micro:bit', replacing regular classes (writing and math) with programming.
The time needed to test and evaluate the CT lessons made in the Novum project, was also incorporated in regular lessons. The involved teachers together with school management, decided that these lessons could be part of the regular school week, and with very good results.
All the teachers, the ones involved in the project and the ones who were asked to participate, were enthusiastic, and learned meaning and the importance of CT. One of the teachers said: "Pupils learn to organise, divide, select and be very precise". That is no standard part of lessons for math and languages. For the pupils these were lessons with new materials, new subjects and a complete different way to learn things. And with great results (e.g. a computer made some songs, after pushing a button). They were very enthusiastic and even wanted to have more lessons like this. Even after they experienced that their Dutch languages had to be completely correct, otherwise the 'computer' could not read it. They have to be very accurate in using their refreshable Braille display, in a fun way. These are some of the benefits of CT lessons. All the teachers did not

experience any extra workload for these lessons. All the pupils participated with great joy and enthusiasm, and we believe that it will be not different for this research.

**Overige (maximaal 7500 karakters)**