The perception of teachers on usability and accessibility of programming materials for children with visual impairments

Meulen, A.N. van der; Hartendorp, M.; Voorn, W.; Hermans, F.F.J.

The Perception of Teachers on Usability and Accessibility of Programming Materials for Children with Visual Impairments

ANNA VAN DER MEULEN, University of Leiden
MIJKE HARTENDORP, Saxion University of Applied Sciences
WENDY VOORN, Royal Dutch Visio
FELIENNE HERMANS, University of Leiden and Vrije Universiteit Amsterdam

Programming education is strongly emerging in elementary and high school. Diversity and inclusion are important topics, however, insights on suited programming materials for younger learners with visual impairments are lacking. A wide range of programming materials for children exists, diverse in both what is being programmed (output) and how this is done (input), yet often relying on visual features. An understanding of the usability and accessibility aspects of these different materials is important to inform educational practice and to increase understanding of what makes programming materials suited for low vision and blind children. The aim of this study is to explore the usability and accessibility of programming materials currently used in education to low vision and blind children in the Netherlands. A focus group was conducted with six teachers or IT experts, all working with the target group in special education. The thematic analysis of the discussion of 25 materials (including unplugged lessons, robots and robotic kits, block-based and text-based languages) showed the potential of several materials, especially unplugged lessons, and the continuing search for suited materials and workforms specifically for the blind children. Furthermore, prioritizing "fun" and close connections to children's daily life as well as careful explorations of usability at the cognitive level came forward as important factors for future research and development in programming materials for low vision and blind children. These insights can contribute to obtaining an inclusive approach to programming for young learners.

CCS Concepts: • **Social and professional topics → K-12 education**; **People with disabilities;**

Additional Key Words and Phrases: Children's programming materials, usability and accessibility, visual impairments

ACM Reference format:
Anna van der Meulen, Mijke Hartendorp, Wendy Voorn, and Felienne Hermans. 2022. The Perception of Teachers on Usability and Accessibility of Programming Materials for Children with Visual Impairments. *ACM Trans. Comput. Educ.* 23, 1, Article 14 (December 2022), 21 pages.
https://doi.org/10.1145/3561391

Authors' addresses: A. van der Meulen, University of Leiden, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands; email: a.n.van.der.meulen@liacs.leidenuniv.nl; M. Hartendorp, Saxion University of Applied Sciences, Handelskade 75, 7417 DH Deventer, The Netherlands; email: m.o.hartendorp@saxion.nl; W. Voorn, Royal Dutch Vision, Amersfoortsestraatweg 180, 1272 RR Huizen, The Netherlands; email: w.voorn@auris.nl; F. Hermans, University of Leiden, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands and Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam; email: f.f.j.hermans@vu.nl.

1946-6226/2022/12-ART14
https://doi.org/10.1145/3561391

1 INTRODUCTION

Programming is being increasingly frequently taught in both elementary and high school, resulting in the development of tools and kits, which can generally be referred to as "programming materials", targeting this population. Diversity and inclusion have become important topics in this trend. Although this includes learners with visual impairments [18, 33], specifically younger low vision and blind children remain underrepresented [22]. Most research has focused on possibilities and challenges in text-based programming, relevant for the older learners of this group at the high school and undergraduate level [48]. Attention is starting to be paid to the younger learners (at the elementary school level, up to around 12 years of age) in the form of exploring the possibilities for children with visual impairments of popular educational programming tools such as the Bee-bot or Lego-tools [22], tangible environments in general [38] as well as the development of promising accessible tools (see for example [32, 33]). Most previous research however targets a specific material, whereas a wide range of often different materials are available and currently being applied in educational practice [56]. A broad exploration of the usability of this range of diverse materials directly from the experience of their use in the context of educational practice, can serve two main purposes. First, it can facilitate insights into how the currently in use materials promote or hinder inclusion within their application in educational practice. This can be helpful for future research on and development of usable materials. Second, it provides directly applicable suggestions for teachers working with a child with a visual impairment who have such materials available.

The aim of this study is to explore the experiences and perceptions of teachers in the Netherlands on the usability and accessibility of programming materials, currently used in education for children with visual impairments. The diversity of the group of children with visual impairments, the wide range of materials available, as well as the current state of programming education (having been on the rise for a while yet continuing to be in an exploring phase, without a fixed curriculum being available) together forms a complex topic which asks for different insights to investigate and build on. Teachers working with low vision and blind children have unique and comprehensive insights into the suitedness of programming materials for these children within the context of their daily educational practice. Consequently, an open discussion of experiences of teachers in the form of a focus group can be a valuable contribute to generate views and ideas directly from practice, that can add to both further research and development and the practice itself. Our research question is: What are the experiences and perceptions of teachers in special education on the use of current programming materials for children with visual impairments?

A final note concerns our language use in referring to the learners of our interest. An ongoing debate exists on appropriate language use when referring to individuals with impairments [46]. Through our manuscript, we follow what is common in the context of our educational practice as well as in line with the academic literature we build on. We speak consistently of "learners or children with visual impairments" and, when applicable, "blind and/or low vision children or learners".

2 RELATED WORKS

2.1 Programming Tools for Children

Programming education to children as currently being taught from as early as kindergarten can take diverse forms. A wide range of tools in the form of lesson plans, assignments, and materials of a different nature are available [56], many of which are currently being used in programming education [23]. Lessons and materials have been designed to be specifically accessible or attractive to children [33, 56], ranging from unplugged lessons where children program each other or the teachers as robots [15], to tangible robots, tools or games such as the Bee-bot or Cubetto, PC or mobile device applications with visual or block-based programming languages such as Scratch or hybrid approaches where physical and virtual parts are combined [56] for example in the micro:bit [44]. In addition, older children are also being introduced to or further taught in programming with regular text-based environments and languages such as Python, JavaScript, or HTML. For some of these tools, different input (referring in this context to all operations of programming the material) and output (referring to the outcome of these operations) combinations are possible, the tangible micro:bit for instance can be programmed with either a block-based language or JavaScript, and Scratch can be used to create stories, games, and animations. Some of the tangible kits or tools also differ in the extent to which they can be assembled or disassembled or whether they have a display. Currently, the most common for elementary school children are visual or block-based programming as input, and graphics or robots as output, yet several different types of input and output are in use [23].

Core features of visual or block-based programming tools (Scratch, Blockly, Alice) are their puzzle-piece-like blocks that represent units of code, the easy code construction through direct drag and drop manipulation, the fact that the blocks only snap together if this is grammatically correct (which reduces syntactic errors), the visible nesting of blocks that indicate the scope, and the close tie between the code environment and code output that facilitate easy understanding of cause and effect [32, 33]. Block-based programming environments have been shown to be valuable tools in programming education, with students working efficiently and concentrated on assignments [39] and learning important computer science concepts [31]. Similar positive effects have been shown for unplugged activities and tangible tools, especially for younger children unplugged being most effective in achieving an understanding of computer science concepts and tangible tools being most engaging [55]. These insights show the potential of the diverse tools available, and their possible different strengths.

2.2 Usability and Accessibility of Technology Use

In order to understand and consequently improve the potential of the types of programming tools described above for specifically low vision and blind children, the usability and accessibility of these tools should be assessed. These two concepts are widely applied within the topic of technology use, including the use of computers in general and specifically programming materials [40].

Generally, usability and accessibility apply to the understanding of the use of products or services. The evaluation of usability and accessibility is aimed at making usable and useful technologies, for all populations [36]. The focus lies on the technology that is used and the context in which it is used, with at the background cognitive processes such as perceiving and processing information through learning, and the use of memory and attention in doing this [12, 35]. Concretely, usability concerns the extent to which a product or service is easily and intuitively understood and navigated through, rooted in the operational experience of the user [35, 40]. It includes the following components:

— Effectiveness: effectiveness refers to the extent to which goals are reached with accurateness and completeness, including the "quality of solution" and error rate [16].
— Efficiency: efficiency concerns the relation between accuracy and completeness with which goals are achieved as well as the resources used to accomplish this.
— Satisfaction: the more qualitative component of satisfaction involves the user's overall comfort and positive attitude and generally refers to well-being or fun during its use [16, 40].

The definition and fundamental attributes of usability, though originated in research with adults, can be easily transferred to the target group of children [35]. Accessibility concerns specifically the extent to which products or services can be used by as many individuals as possible, and is, though not exclusively, often applied to the situation of individuals with impairments [40]. Accessible products are perceivable: all users are able to perceive the presented information; operable: all users should be able to operate the interface; understandable: contents and operations should be within the users' ability to understand; robust: accessing the content should remain possible as technologies advance (Web Content Accessibility guidelines).[1] What this means in the context of programming materials for low vision and blind learners becomes clear in the description of some of the known issues below. Various accessibility issues are known from studies into computer use in general by individuals with visual impairments, including the role of complex factors such as changes in vision and social issues [45, 51]. The latter refers to the possible experience of the importance to blend in, and the resulting reluctance to use visible assistive technologies [51].

2.3 Programming Materials and Children with Visual Impairments

Recently studies have started to focus on programming education for younger learners with visual impairments, often by looking at a specific material, for example the Bee-bot [22], or a specific type of programming such as block-based languages [18, 33]. Overall it is being stressed that both the input and output of materials for children rely heavily on visual features, in order to make them suited and attractive for children [33]. Consequently, many materials, especially the block-based environments, are seen as not accessible for learners with visual impairments [32, 33]. Confirmed usability issues are the drag-and-drop interface, the reliance upon the mouse, and the output in the form of animations [18, 33]. Other types of materials with tangible inputs or outputs, Bee-bot, Cubetto, and buildable robots such as Lego WeDo, have been proposed or confirmed (in the case of the Bee-bot, [22]) to be better suited, and teachers are findings ways to compensate for the inaccessibility of block-based environments [34]. However, many challenges remain. Tangible materials often contain visual properties to indicate the difference between pieces and how to connect them as well as in the experience of the outcome [33]. The Lego blocks of Lego WeDo and Lego Mindstorms for instance are small and difficult to handle [24, 28]. Finally, issues in text-based programming are known by older learners and include the use of screenreaders in the programming environment and language, and problems with navigating through code [1, 2, 18]. It remains open though whether these same issues apply to the situation of younger learners with visual impairments using text programming. Because of young children's overall developmental level and their phase of programming education, other usability issues exist. For all non-English younger learners for example English text-based languages can be a problem.

Several initiatives have started over the past few years to adapt or design materials aiming to make them suited for children with visual impairments. This concerns first and foremost block-based (types of) programming, either by improving the suitability of existing environments [27, 32] or by designing new tools [33]. These different adaptations and tools include accessible touch screen block-based environments (Blocks4Al [32]), applications to work with screenreaders in block-based environments (StoryBlocks, [27]), Accessible Blockly [28], and tangible block languages (Torino/CodeJumper [33]). They have in common that they build upon the advantages of block-based environments [32, 33] and address specific issues for children with visual impairments identified in these environments as specified above. In the case of Lego Mindstorm Robots, both the input in the form of a block-based environment and the graphical output have been adapted for low vision and blind children, using adaptations such as a tray to organize building parts, written instead of depicted instructions and a text instead of block-based language [28]. In the case of text-based programming adaptations include the inclusive programming language Quorum [48]), an Audio Programming Language [45], and tools that provide audio feedback when navigating through code [5, 49].

[1] https://www.w3.org/TR/WCAG21/.

2.4 Accessibility at the Cognitive Level

Finally, a less explored topic within children with visual impairments' use of programming materials is their conceptual development and mental modeling. It has been pointed out that usable and accessible technologies involve more than for instance the addition of audio to a visual environment, and should take into account considerations at the cognitive level of experience and understanding [45]. Learning programming involves specific cognitive processes and abstract concepts. These abstract concepts can be perceived as difficult to begin with [43], but moreover adequately conveying abstract cognitive concepts to learners with visual impairments is known from other educational fields such as science [26] and music education [4] to be especially challenging. This is a complex multifaceted area, but some important considerations can be first of all that concretely materials or lessons themselves can make use of certain concepts that are primarily visual. A focus group with high school teachers for low vision and blind students pointed out that in unplugged lessons visual metaphors can be a problem [48]. When such metaphors are made accessible they can be understood by learners with visual impairments, but with more difficulty. Second, issues can arise since concepts such as abstraction are related to the development of visio-spatial mental modeling and spatial navigation, areas of development where blind individuals can have certain specificities [52, 54]. In early programming education, more complex and abstract tasks increasingly involve moving away from direct manipulation control (moving a robot step-by-step by hand across a maze) and toward computational control (constructing a set of instructions and executing it later on) [25]. Consequently, skills of mental representation and modeling of the assignment and environment are always involved, yet become increasingly important when cognitive demands of the task increase [25]. Taken together, the experience at the cognitive level of abstract concepts and mental modeling for learners with visual impairments is important to explore further, also for specific materials.

2.5 Summary and Current Research Overview

Together, previous insights and initiatives provided an understanding of several usability issues and potential solutions, within different aspects of the input and output within the range of materials available for children. However, currently, many different types of materials are being applied in education [23, 56] and insight into what is being explored for children with visual impairments and how this is experienced is lacking. For several reasons such insight into this complex topic is important. First of all, the majority of children take part in regular education. Although materials specifically designed to be suited for low vision and blind children can be highly valuable options, it should be taken into account that the understanding and improvement of most commonly known and (in regular education) most often applied materials is essential as well to achieve inclusive programming education. Second, individuals with visual impairments form a diverse heterogeneous group [9, 32] with most individuals having low vision and a smaller group being completely blind. In addition to different individual possibilities, experiences, and preferences in the practical use of technology [9], differences in mental or spatial models can have a role in this topic, as well as social issues of wanting to blend in with classmates [51]. Consequently, a valuable addition is the perspective of teachers, an until now underrepresented element [21]. Teachers, especially those in special education, can consider this topic from their comprehensive experience of educating low vision and blind children. Also in a complex topic such as individuals with visual impairments' mental modeling, the experience of teachers in conveying abstract concepts can be very insightful.

In order to gain insight into teachers' experience and perception with current programming materials in use in the context of educational practice, a focus group with teachers is a suited approach. A focus group is aimed at gaining insights into perspectives based on personal everyday

Table 1. Participants of Focus Group

Participant	Age	Sex	Occupation
T1	29	M	Teacher, IT coordinator
T2	33	M	Teacher
T3	–	V	Teacher, IT advisor, and coordinator
T4	41	M	Teaching assistant, teaching support
T5	35	V	Teacher
T6	–	M	Teacher, trainer, IT advisor

experiences, from a selected group of individuals [17, 50]. It can be especially suited to explore opinions and perspectives on a complex topic, making use of the dynamic nature of the discussion and the interaction between participants [17]. This method fits the purpose of gaining a practically based understanding of how teachers at the moment experience the usability and accessibility of programming materials for low vision and blind children. In addition to exploring their perception on established basic elements of usability and accessibility, additional contextual ideas can be generated on what is most noticeable and relevant at the moment when using programming materials in the context of the classroom of children with visual impairments. This can be valuable both for practitioners and for future research on the topic.

3 METHOD

3.1 Participants

The six participants of the focus group all worked in special education schools for children with visual impairments, as teacher, teaching assistant, and/or IT expert or advisor. Two of the participants were female, the other four male, and their age ranged from 29 to 41 years old (two teachers did not provide their age). Table 1 summarizes the background of the six participants. The teachers worked at three different schools which were all part of one of the two Dutch expertise centers for people with visual impairments. We selected the teachers based on their experience with teaching programming to children with visual impairments, in their own classess or during workshops or thematic events. Other specific programming experience or expertise in programming was not a requirement for the teachers. The expertise center of which these teachers were part of has five schools (one of which only includes children who also have cognitive impairments). We approached the other Dutch expertise center as well, however, here teachers indicated to have much less experience exploring programming for younger children.

The focus group discussion concerned children with visual impairments (low vision or blind), ranging in age from kindergarten till the end of high school, with a focus on children at the elementary school level. We included teachers from special education since they have the most experience with our specific target group. Consequently, the children these teachers work with form a diverse group with additional special education needs. The latter is not the focus of our topic but can be kept in mind when interpreting our findings.

3.2 Procedure

One of the research team members, who worked at the Dutch expertise center, recruited the participants through their network, contacting them through e-mail. Participants received the informed consent letter beforehand and were asked to inform the researcher which programming materials they had worked with together with the children, in order for the researcher to have an overview of the programming materials that would be discussed. We asked the participants to bring these materials with them. The informed consent letter explained that the aim of the focus group was to gain insight into the interaction of children with visual impairments with programming materials. Pseudonyms were used in the transcript of the focus group.

The set-up of the focus group, including the selection of participants and topic list as described below, was aimed at making use of this method to draw personal daily life experiences and moreover explore these in an interactive discussion [17, 50]. The discussion took place in one session, organized at the office of the expertise center. The participants were first given some time to read through the informed consent letter and ask questions, and asked to fill out the informed consent form. The researcher was present as a facilitator and sat together with the participants around a table on which all the programming materials brought by the participants and researcher were laid out. A camera was placed to record the group. The focus group lasted 1.5 hours.

3.3 Focus Group

3.3.1 Focus Group Method and Protocol. Because of the open and exploratory nature of the research, any form of an unplugged or plugged tool, artifact, environment, or language was considered a programming material and could therefore be discussed. The term programming material was not defined by the teachers. However, the starting point was to only discuss materials at least one of the teachers had worked with themselves together with the target group.

A protocol for the focus group was designed in which six main topics, concerning usability and accessibility of the materials, were distinguished. These topics were:

- Experience with and characteristics of the material, for example, was a material used in a lesson series, or in an introductory workshop.
- Concrete usability of the material, for example, how can children handle the different aspects, primarily the input and output, of the material.
- Working with the material, for example, can children start an assignment with the material, and how much support is need.
- Workforms the material invites, for example, do children collaborate, do they start to explore and add their own creativity.
- Presence or absence of fun and comfort, for example, do children enjoy working with the material, do they get motivated, is it frustrating, is it boring.
- Differences between the children, for example, can all children work equally well with the different aspects of the materials.

After the introduction, the facilitator asked teachers whether there was a material they would like to start with. The role of the facilitator was to monitor that the different topics were addressed for each material and that all materials were discussed, to ask for further clarification or confirmation, and to encourage all teachers to participate and to generally facilitate the dynamic of the session. There was no specific order in which the materials or topics were discussed; the course of the conversation was followed. During the reflection and closing section, the teachers were asked to pick a favorite material among the materials discussed and were given space to add other comments. The materials that were present were not used during the focus group but when a specific material was discussed it was occasionally pointed at or picked up, or different aspects were pointed out on the material.

3.3.2 Transcription and Thematic Analysis. The focus group was transcribed verbatim from the video. We followed a thematic approach to the data analysis, generating themes from the data related to the interest and topic of the research aim [8, 10]. Both the transcription and thematic analysis were performed manually. First, a thorough inspection of the transcript was performed, categorizing the transcript into different sections in line with themes that emerged and labeling each section with this theme. This was done in two separate ways: based on specific programming materials, for instance, labeling all information on the Bee-bot with the label "Bee-bot", and using a more open approach where labels directly related to features of usability and accessibility, for example "working independently", "intense guidance by the teacher", "positive experience material", were added to the data. Continuing with the open approach, these labels were gathered under broader themes ("working independently" and "intense guidance by the teacher" were gathered under the theme "workforms"). All relevant information in the transcript received a label, making sure that all information was considered in the next steps. Information was only considered non-relevant if it did not relate to programming education of low vision and blind children. A second assessor was involved to check the approach of labeling the information from the transcript. The second assessor independently labeled three excerpts from the transcript, after which the first and second assessor compared their labels. In several cases, the labels used by the two reviewers were entirely similar (for example "positive experience" and "difficult for blind children"). Other labels differed not in the content but in for instance the level of detail (with one assessor distinguishing two specific detailed types of labels whereas the other assessor took this together as one). For such cases, the assessors discussed until a consensus was reached. The overall themes emerging from gathering the labels were also discussed. Finally, the two types of themes (specific materials and usability/accessibility features) were taken together, using the themes that emerged from the open thematic analysis as a starting point and complementing them with material-specific information from the overview per material. Quotes are used to illustrate an overall view as described for a theme, or to indicate a specific opinion. Since data derived from a focus group are interactional in nature [47], a reflection of the group dynamics and atmosphere is helpful to interpret the insights correctly. Consequently, the final section of the results describes the overall group process and some specific observations. The focus group was conducted in Dutch. Quotes provided in the results section have been translated into English.

4 RESULTS

4.1 Overview Materials and Themes

In total 25 materials were discussed. Materials were included based on the teachers' experience. Table 2 summarizes all materials including their type and whether there are additional parts to the material. For those materials categorized as "hybrid", we also indicate whether the input is virtual or tangible. The experience the teachers had in using the materials with the children differed strongly, however with most materials at least two teachers had more than an introductory experience. The teachers had above-average experience working with the children with the micro:bit, Bee-bot, Blue-bot, Scratch, and all four unplugged materials, and they had not worked at all with the children with the Arduino, Python, Tickle, and Sphero.

The open thematic analysis resulted in the following six main themes, which are described in more detail below:

— The search for options for blind children/options for low vision children
— Work set ups (collaboration, independence, and instruction)
— Battery of consecutive (types of) programming materials
— Working indirectly with the input
— Transfer and comprehension of the way of thinking and concepts (cognition)
— Subjective experiences of satisfaction, well-being, and fun

4.2 Theme 1: Search for Options for Blind Children/Options for Low Vision Children

One of the most central and recurring themes in how the children interact with the materials was the difference between blind and low vision children, mostly because of issues and lack of options

Table 2. Overview of Programming Materials Discussed during the Focus Group

Material	Type	Type of input if hybrid	Combined with
Alive Programming	physical not electronics	–	–
Sandwich Robot	physical not electronics	–	–
Robotdance	physical not electronics	–	–
Hello Ruby	physical not electronics	–	lesson kit
Cubetto	physical electronics	–	–
Bee-bot	physical electronics	–	wooden maze
Sphero	physical electronics/hybrid	virtual	programmed by Tickle app
Blue-bot	hybrid	tangible	wooden maze, tactile reader, cards
micro:bit	hybrid	virtual	MakeCode environment
Fabel	hybrid	virtual	programmed with block language
Cyberrobot	hybrid	virtual	programmed with app on ipad
Maki Maki	hybrid	virtual	programmed with Scratch
Nao	hybrid	virtual	(input not discussed)
Lego Boost	hybrid	virtual	app on IPad
Lego WeDo	hybrid	virtual	app on IPad
Snap Circuits	hybrid	virtual	app on IPad
Mambo Drone	hybrid	virtual	programmed with Swift Playgrounds
Arduino	hybrid	virtual	programmed with JavaScript
Swift Playgrounds	virtual/hybrid	–	used to program Mambo Drone
Tickle	virtual/hybrid	–	used to program Sphero
Scratch	virtual	–	used with Maki Maki
HTML	virtual	–	–
JavaScript	virtual	–	used with micro:bit
Sonic Pi	virtual	–	–
Python	virtual	–	–

the teachers experienced for the blind children. The first four teachers especially expressed their struggle to find materials equally suited for both low vision and blind children, referring to this as a continuing search. As one of the most experienced teachers indicated: "Well we had the NAO in the class the other day, and then you hear some blind children say, yeah, yeah, I do not understand what was happening on the board. So you are searching, how can you involve them with that, how can you provide the insights, is this possible, and if yes, how?"

As the example indicates, several materials are less or not at all usable for blind children, often because of the input. For example, the micro:bit, NAO and Fabel can as outputs be tactilely explored by blind children, who can trace their movements. The programming itself however, using block-based environments such as the MakeCode environment for the micro:bit, cannot be followed specifically by the blind children while according to the teachers for low vision children this can work. For other materials both the input and output are difficult for blind children. Manuals on the iPad for the Lego Boost, Lego WeDo and Snap Circuits are very visual and complex, and adapted forms of the manuals can still only be used by low vision children. The output of the Lego materials, as well as the Cyberrobot, are also not usable for blind children because the parts are too small. Snap Circuits are better usable by blind children because the parts can be easily distinguished tactilely and the pieces are placed in a frame.

Unplugged materials are according to the teachers the only type of tools with which blind and low vision children can participate equally. Several teachers expressed their enthusiasm about these tools, stressing the whole class can be involved actively and equally: "It is just really appealing. It speaks for itself actually, they all have an idea of it. Blind or low vision, it does not matter".

4.3 Theme 2: Work Set Ups: Collaboration, Independence, and Instruction

The teachers talked about the work set-ups they used in working with the materials, often related to the options and obstacles in terms of the usability of the materials. First, teachers frequently work in a frontal setting (meaning the teacher addresses the whole class together), with the whole class working together and guided by the teacher. The teacher for example programs on the Digiboard while the class works on the code together (using Scratch or a different block code), programming different robots as outputs (Fabel, NAO). Other robots are constructed jointly by the class (Cyberrobot). The Cubetto was explored in a set-up where the teacher interacted with the material and the children watched and thought along. Second, collaborations between low vision and blind children are brought forward multiple times as an approach to work around the lack of usability for blind children. With Snap Circuits the low vision child can follow the manual, while the blind child can build the output because the blocks and frame of Snap Circuits are tactilely accessible. With Swift Playgrounds the low vision child can notice typing errors in the code. One teacher is enthusiastic about this approach, since children enjoy it and experience it as a challenge to complete an assignment together, but two other teachers prefer that all children can work independently. Furthermore, it can be seen that blind children become in this set-up passive and lose interest: "And just sitting, waiting until your buddy says something, about what's going on, that makes them very inactive, and then the interest disappears actually very fast, I have noticed". Interestingly also one example is given of a duo working with the micro:bit where the blind child is more advanced and consequently takes the lead. This seems in line with a discussion elsewhere where several teachers agree that there are certainly some blind children with an affinity with programming who transition well toward text programming, after which they can make good progress.

Furthermore, independently working (individually, but also in pairs or groups without constant guidance from the teacher) is seen as desirable but at this point difficult. The teachers mention the Bee-bot and Blue-bot as the only materials with which currently both blind and low vision children can work by themselves. With little instruction pairs of children can start to explore or play together. This works well because both the Bee-bot and the wooden maze are tangible, and the tactile reader for the Blue-bot provides a logical frame suited for blind children to put the cards in. Low vision children can work independently with other materials as well, once they have learned the basics. With the micro:bit, Scratch, and Sonic Pi children can work by adding their own ideas and variation starting from a step-by-step plan, or by starting with a worked out example and continuing on their own. By themselves, they can explore and improve their code. According to the teachers, enthusiasm can be seen in the children because they are in charge of their own creation, also contributing to sharing work with each other and proceeding in collaborations. "That is really fun, you do the step-by-step plan (in Scratch), and that they then for example a different figure, they say yeah, can I have a different figure. Yes of course, try, go ahead make a new figure. That is then possible", "And then I have, that one blind student worked with the program (Sonic Pi), and the other I gave a keyboard, and that one made a melody at the keyboard, and then says which tone that is, and then the blind typed that, and then trying, is that correct, no, not correct...they really independently made something, a song".

Finally, overall the teachers also experience that more effort, instruction, and supervision are needed for blind children. This is also the case for a material such as the Bee-bot. Even though eventually blind children are able to work with the Bee-bot by themselves, extra context is needed to explain the idea of the robot and extra time to explore the robot, the buttons, and the maze. The teachers are searching for ways to give suited attention to the blind child without holding back the low vision child, giving more intensive support to the blind child, providing more time and repeated instructions, different assignments, and extended introductions and examples.

4.4 Theme 3: Battery of Consecutive (Types of) Programming Materials

Two teachers summarize the different possibilities of types of materials by placing these materials in a "programming materials battery", which other teachers confirm. This battery consists of three or four steps in learning programming, with each step being related to a specific type of materials. The first step is unplugged, followed by block programming (possibly divided into tactile block programming and simple (“baby”) text programming), and finally by actual text programming. As mentioned, the teachers are very happy with the possibilities for unplugged, with which almost all of them have worked. The lessons work well to illustrate how a computer works, to teach thinking in steps, and identifying mistakes in these steps. One teacher explains how the unplugged lessons start valuable discussions: “and so you get into really fun situations, that you just talk with them [the children] about how dumb a computer actually is, right, and how smart we are, that we just get that step right away and that a computer has to do that all step-by-step, and as soon as there is just one mistake then it does not work anymore”. Furthermore, the lessons have humor, keep the children actively engaged, connect to their daily life, and most importantly allow for equal participation of blind and low vision children. The step of block programming however is experienced as entirely absent for blind children, which is related to the previously mentioned search for usable materials.

The teachers stress the importance of block-based possibilities or alternatives for blind children because of the clear advantages: it builds on the level of unplugged, allows for trying out code and receiving immediate feedback (in the form of blocks being able to click together) and for the children to work by themselves. Whereas the next step of text programming has some possibilities for blind children (for example working with HTML), specific environments (MakeCode, Sonic Pi) are not usable with screenreaders. Moreover, text languages are complex because of the length of the commands and the need to be exact in spelling and punctuation. For younger children (middle of elementary school) this is not suited, and also not all older children get to this step, especially when there are no options for the previous block-coding step.

4.5 Theme 4: Working Indirectly with the Input

Some materials are accessible in their output but not their input. An example is the micro:bit, which the teachers see as their favorite material for older children. The output of the micro:bit is suitable and fun for both low vision and blind children, mostly because of its physical nature and closeness to children’s daily life (an alarm clock, or Christmas ornament can be created). The input, however, writing either block-code or text code (JavaScript) in the MakeCode environment is (with the block-code option) only usable for the low vision children. Teachers describe different ways to use a material for which the input is not accessible while the output is.

First, the teacher him or herself can do the actual coding with the material, while blind children use braille to describe the code outside of the material. One teacher used this method to program the micro:bit with block language in the MakeCode environment. A blind child wrote the code down in braille, and the teacher converted this exactly as the child had written it in blocks. Another teacher also used braille working with Swift Playground, where text coding is used. The teacher made an overview of functions in braille, which the children could select. Second, inaccessible input can be made usable by adding a text processor (compatible with assistive technologies the children use, such as a screenreader) next to the code environment. This method was used as well by one teacher in the MakeCode environment of the micro-bit but then using JavaScript. A blind child wrote code in Word and copied it to MakeCode. This teacher explored the same approach with Sonic Pi, with a blind child writing code in Word and copy-pasting back and forth between Word and the Sonic Pi environment to run and adjust the code.

The teachers are hesitant about especially the approach with braille with the micro:bit, which has the advantage that it makes it possible for blind children to work with the output, to write down the steps themselves, and to complete an assignment but has the disadvantage that the children don’t work directly with the software and need intense guidance. One teacher, who was not familiar with the setup of Sonic Pi and Word, became very enthusiastic about this option after hearing about it from the other teacher.

4.6 Theme 5: Transfer and Comprehension of Way of Thinking and Concepts (Cognition)

Several times through the discussion the teachers refer to how the different materials can contribute to, or show, children's understanding of the reasoning and concepts of programming education. The teachers agreed that especially unplugged lessons can work very well in clarifying "the steps and logical thinking" and understanding of how a computer works: "it works exact, that you have to be exactly clear, indicate step-by-step. It is perceived as natural and consequently it really sticks, and they enjoy it". It is also indicated that even when children cannot directly work with a material, they can learn about programming by discussing, or writing down, the steps of a program. The teachers expressed that they had been positively surprised by the understanding the children showed in logical reasoning and grasping the connection between actions.

In this topic as well, challenges for the blind children were pointed out. Two of the teachers referred to an issue in conveying the logic of programming to the blind children. Partly, they described this in the context of inaccessible or less easily accessible input and output: blind children cannot be involved sufficiently in discussing the programming steps when working with a block-based language they cannot see. The basic idea and logic of a plan for the Bee-bot to move toward a destination is also less directly understood by blind children and requires a step-by-step tactile exploration of the environment. In addition to these issues, one teacher also considered whether blind children are less familiar with thinking in steps, to begin with (as he observed when working with the Bee-bots). In the discussion on unplugged materials, however, this same teacher mentioned he thought blind children are in fact more used to this step-by-step thinking from their daily life. The other teachers appeared more in agreement with the latter idea: "but I also think that a blind student thinks more about these kinds of things him or herself. A low vision sits here and then goes, oh, I go to the door, and walks automatically, oh there is the door and there I go too. Whereas a blind student has to think first, okay, then first I will look for.., then first I will think where the door is. Then he goes, he goes that way (teacher turns) and then feels..". Finally, there were mentions of concepts used in materials or assignments that are very visual in nature and hard to adapt and/or explain. This included "making half a turn" or "turn toward the window" in unplugged lessons, and the arrows on the Blue-bot cards. Apart from the visual nature of these arrows, according to the teachers, it is difficult to understand for blind children how an arrow works, and consequently, a tangible version of these cards is not a solution. The teachers are experimenting with either L and R for left and right in braille on the cards or alternative arrow-shapes.

4.7 Theme 6: Subjective Experiences of Satisfaction, Well-being, and Fun

Although as the themes above show the teachers identify several challenges and current problems in their experiences with programming materials for blind and low vision children, overall they also emphasize both their own and the children's enthusiasm. According to the teachers, children enjoy discovering the connection between input and output, and output in forms close to their daily life. They also see that the children like the experience of autonomy when they create something themselves, the creative and active way they can work with the materials, as well as the humor in some of the unplugged materials and the novelty of some of the robots. One teacher summarized: "..I think that they are, they are about the programming lessons, in my group very enthusiastic. It almost does not matter what I use."

Blind children are a little bit less automatically enthusiastic and involved, and do not like it when they cannot follow what is happening such as with block-coding on the Digiboard. The children in general do not like it when materials are too difficult or when the connection between input and output is too slow (it takes too long to experience the effect of what they are doing) which is frustrating for them, when they cannot actively work themselves, and when they are not able to trace an error.

Though teachers overall agree on children's experience, there was also some discussion about the value of specific materials. Snap Circuits and the Lego materials were appreciated by two teachers whereas one other teacher (one of the most experienced) considered them clearly unsuited both because of their input and output. The teachers differed in opinion as well on the extent to which certain materials have the option for more advanced learning and more depth, most clearly concerning the Bee-bot. The Bee-bot was picked as the favorite material for the younger children, but especially one teacher was somewhat skeptical about its potential as more than a toy. For the older children, the micro:bit was picked as a favorite, however only for the low vision children. The unplugged lessons were also seen as a favorite overall. Finally, one teacher was explicit in that it was difficult to pick a favorite material, emphasizing both that so many materials worked well and that actually none could be picked that was equally usable by blind and low vision children: "So I'll leave it a bit open. I am still a bit like, this..especially if I look at, if I look at the low vision then I say micro:bit. But I, I look, actually at the whole group. (...) Yes, because I think this is the best, if you can really do it all together".

4.8 General Atmosphere and Individual Contributions

Through the focus-group discussion, all participants appeared involved and willing to share their ideas and opinions, though some teachers (teachers 2 and 4) were more active than others (teacher 5). Overall the teachers had quite strong opinions and ideas and appeared honest, disagreeing with each other several times and occasionally indicating they just did not have an answer to a certain question.

Teachers 2 and 4 were most strongly opinionated, but differed in how certain their ideas were. Teacher 2 was contemplative and mostly seemed to think aloud when expressing his ideas (for instance on the importance of being able to involve all children equally) while teacher 4 stood out because of the certainty with which he expressed his ideas. Teachers 1, 3, and 5 all stayed somewhat closer to sharing their own concrete experiences, but at times were also open to reflecting on other ideas, either remaining doubtful (about the potential of certain robots) or becoming enthusiastic about the new ideas or options they had not considered themselves. Teacher 6 appeared to have the least classroom experience but shared several examples of individual experiences with children. Finally, most of the teachers knew each other relatively well, and were already together involved in the topic. On the one hand, this appeared to contribute to a quite easy flow of conversation, with teachers being able to occasionally refer to specific students as examples. On the other hand, some of the experiences were similar, lessening somewhat the sharing and emerging of new ideas. It appeared helpful that the teachers had worked with different age groups and had often also tried out different materials or the same materials in different set ups.

5 DISCUSSION

The aim of this study was to explore how teachers of children with visual impairments, working in special education in the Netherlands, experience the usability and accessibility of programming materials currently in use in educational practice. The focus group conducted with six participants provides an understanding of how different programming materials are perceived and moreover what teachers consider and prioritize in their usability experience. Below the insights are discussed, highlighting the contribution this perspective of the teachers can make.

5.1 Overall Experience Usability of Input and Output

The teachers explored a wide range of materials with their children, including all types of materials as previously distinguished [56]. This diversity illustrates the stage of "searching for what works" the teachers describe themselves to be in (especially for the blind children), and reflects the general availability and use of materials both for sighed and low vision and blind children at the moment [23, 33, 56]. Furthermore, virtual or hybrid materials are currently most popular in elementary school programming education [23] and were also most employed by the teachers of the focus group. Interestingly however was that relatively few physical electronical and hybrid tangible materials were mentioned by our teachers. Several such materials exist [56] which seem intuitively suited for the target group. It is possible the teachers' overall positive experience of the Bee-bot and Blue-bot made them less inclined to try out other comparable tools.

A clear distinction could be seen in the experience of the usability of the output and input of materials. Concerning the output, teachers were relatively positive about the possibilities for both low vision and blind children. Aspects that contributed here included tangible (Bee-bot, Blue-bot) and auditive (Sonic Pi) features, connections to daily life (micro:bit, unplugged), novelty of robots, the inclusion of humor (unplugged), and the possibility for children to add their own ideas and creativity (Sonic Pi, Scratch). A fast and evident connection between input and output, elsewhere indicated as a "feature of liveness" [33], can be seen in several of these materials and is also especially valuable for low vision and blind children. Some of these outputs have already been identified as suited for children with visual impairments, for instance, the Bee-bot [24]. Most new here was the enthusiasm about unplugged lessons, which is further discussed below. Concerning the input, however, more challenges were perceived. Especially in this topic, the teachers stressed the difference between low vision and blind children, with often the input being not handleable specifically by blind children. Although in the case of block-based programming its lack of suitedness for children with visual impairments have been widely noted [33], interestingly the teachers of the current study were quite enthusiastic and positive about the possibilities for specifically low vision children. This nuance can be valuable because of the known diversity of the group [9] and the resulting requirement of different possibilities for inputs and outputs. Especially for teachers in regular education, where block-based programming is currently the most popular material [23], who encounter a student with low vision it can be good to be aware that block-based programming is likely accessible for this student.

The teachers experienced a problematic lack of options for the blind children at the level of block-based programming, which makes it difficult to transfer to text-based programming without this experience. It might be however that there are options, directly addressing some of the issues for the blind children, the teachers are not aware of. First of all, the teachers speak of the need for "tangible block-based tools", which actually is how the Blue-bot (when used with the reader and card) and Cubetto have been presented [37, 56]. Second, the type of "easy" text-based programming the teachers are looking for might be found in a language such as Quorum, also designed to be inclusive. There might be different reasons why the teachers don't perceive these tools as options. The Blue-bot can be experienced as limited in programming options, and the Cubetto was, based on one exploration, seen as lacking easy or logical use according to one teacher. Several other options for tangible block-based alternatives, designed for children in general [37] and for the specific target group [32, 33] as well as Quorum, might not have reached the teachers yet. Consequently, a practical and comprehensive overview of available programming materials (general and targeted for the specific group of learners with visual impairment) would be highly beneficial. At the same time, since it can be expected that this continues to be an evolving field, it can be thought of how to best keep practitioners informed on new developments.

5.2 Independence and Instruction in Working with the Materials

The level of independence, the possibility of collaboration, and required instruction all are important when assessing the usability of materials within the classroom. First, the teachers considered children's ability to work independently an important feature of programming materials which is currently, especially for the blind children, difficult to achieve. In the case where this is possible, with the Bee-bot or Blue-bot for younger low vision and blind children, and with Scratch, the micro:bit or Sonic Pi for older low vision children, the teachers see that children start exploring, become creative, and feel in charge of their own work. These are all important usability features, with the experience of autonomy being of particular value for the target group [30]. Furthermore, the workforms the teachers employ when independent working is not possible at the moment because of usability issues in the input, include collaborations between a low vision and blind child or between a blind child and the teachers, as well as the use of additional accessible software. For the micro:bit as well as for Sonic Pi children can write their code in a separate, accessible, text editor. These "workaround" methods [1] have been described elsewhere for diverse levels and age groups. The set up of a blind and low vision child collaborating [33], sighted software developers assisting blind software developers [2] and the use of a separate text editor [18] all appear common methods to work around accessibility issues for individuals with visual impairments. The use of an accessible text editor can also be seen in high school students learning programming [7], a case study of a blind university student [11] as well as blind professional software developers [2]. There do not seem to be previous accounts of this method for this age group. Although workaround methods are also according to the teachers not ideal, mainly because of lack of direct interaction with the material and risk of passivity for blind children, depending on the specific situation and value of other features of the material they might be worth considering.

It also seemed teachers were themselves searching for what "being able to work independently" would mean exactly in this context, and what amount of instruction they find appropriate. For Scratch, the micro:bit and Sonic Pi they describe a method where the children can work well themselves with a step-by-step plan, with an example, and after a first instruction, however for other materials (Bee-bot) they find the necessary initial instruction for especially the blind children too elaborate. Both in the context of special education and in the context of the general programming education debate can be found on appropriate instruction [20, 42] in line with the search of our teachers. In special education (not specifically concerning programming lessons), practitioners continue to search for appropriate tailored instruction fitting the unique needs of specific children [42]. Furthermore, in the field of programming education, as well different teaching strategies are being debated, with the traditional method of discovery learning being questioned and direct instruction being presented as an alternative [20]. Evidently, considerations about inclusive programming materials are directly connected to questions on what is realistic and desirable in terms of independence and instruction of both blind and low vision children. Further studies on usability and accessibility could take in a multi-dimensional approach from the start where support and teaching are incorporated [3, 21], combining ideas from the fields of programming education and special education.

5.3 Unplugged Programming and the Impact of "Fun"

Two specific topics for which new ideas were generated by our teachers concern the potential of unplugged tools and the related value of the factor of fun, and considerations for usability at the cognitive level. The latter is discussed in the next section. Concerning unplugged programming, the teachers were highly enthusiastic about this approach for low vison and blind children, identifying it has the only tool with which all children can be equally involved. Unplugged lessons for learners with visual impairments have not received much attention (though see [48] on high school students), perhaps because research for this group often focuses specifically on issues working with electronics. Unplugged programming however is generally presented as an inclusive option for programming education, because of its lack of reliance upon computers and relatively easy adaptability for specific target groups [13]. These features can also be seen in the unplugged tools the teachers described in this study, which either required already accessible or easily adaptable artifacts or none at all. Moreover, unplugged programming is a popular and efficient approach in programming education at the moment [15, 19]. This emphasizes the value of identifying unplugged programming tools as an available and suited option when working with learners with visual impairments in educational practice.

Furthermore, the positive experience of unplugged tools also highlights which features of programming materials can increase usability for children with visual impairments. In addition to unplugged tools being usually easy to handle or adapt, the teachers also stressed the overall experience of the tools as fun and appealing, including humor and a close connecting to children's daily life, and being effective in conveying ideas about computers and logical thinking. "Fun" is in the case of children a central aspect of the user experience [16, 40, 41]. Also within a learning context, it is important that experiences are enjoyable and relate to learners' lives [53]. Specifically having fun while programming positively effects children's attitude toward coding [53]. Although this applies to the general population of (young) learners, it might be all the more relevant in the case of children with visual impairments. Putting these concrete aspects of user satisfaction of having fun, experiencing humor, and a close connection to daily life in the foreground can involve them and counteract some of the more hesitant attitudes described for especially the blind children. One of the teachers of the focus group also nicely illustrated how the enjoyment and natural experience of unplugged tools go hand in hand with their potential to add to the understanding of programming.

5.4 Experience at the Cognitive Level

The experience of the content of programming at the cognitive level is a largely unexplored area in the context of programming education for learners with visual impairments, apart from some suggestions that this level can have a role in usability [45, 48]. Our teachers referred to both the issue of visual metaphors or concepts within programming tools and possible obstacles in cognitive processing. Concerning the first point, the example of the arrow concept on the Blue-bot cards illustrates the complexity of adapting visual concepts since, similar to what has been previously pointed out [48], tactile versions can be helpful to make the shape accessible but do not lessen the complexity of the concept itself. Possibly, insights from fields with similar issues such as science can be helpful [26]. At the same time, however, it is likely that for specific concepts of programming this is a challenging area that requires its own exploration, as the teachers describe to currently undertake for alternative arrow shapes for the Blue-bot cards. Second, the teachers also referred to the understanding of the idea of planning a robot to move through an environment and thinking in steps. Some of the issues here as considered by one of teachers, of grasping the environment and being hindered by the lack of a direct overview as a sighted person has, can be related to mental modeling and spatial navigation challenges. Challenges in this area for blind individuals are known to exist but are highly dependent on specific tasks, situations, and individual factors [52, 54]. Interestingly, our teachers mostly agreed upon the idea that blind children could have some advantages in their cognitive understanding because thinking in steps is more natural to them as they make their way through an environment for instance. Ultimately, these ideas are very tentative (also taking into account that the challenge of understanding a robot moving through an environment was pointed out only by one, strongly opinionated, teacher), but can be seen as interesting starting points to further explore how considerations at this level for children with visual impairments could concretely become apparent in the practice of working with different materials. For instance, it can be investigated what sort of environment for a Bee-bot or Blue-bot can be most easily grasped or mentally represented by blind children. From the perspective of evaluating materials it is also worthwhile to note that the possible advantage the teachers described of blind children's ability to think in steps might be activated especially in the sort of assignments that connect well to their daily life experiences, stressing again the value of this feature.

5.5 Takeaways for Educational Practice

Finally, some practical suggestions can be given for those working with low vision or blind children on programming education, concerning both the choice for and way to work with materials.

— A lot of potentials can be found in the currently available programming materials to be applied in education for children with visual impairments, with some materials being quite readily available to be used (unplugged lessons, Bee-bot for the low vision and blind children, block-based programming and the micro:bit for some of the low vision children).
— When materials are not (completely) usable, work-around-methods can be considered as the next best thing. These can include collaborations between low vision and blind children, as well as the use of word processing software compatible with a screenreader to write the code in.
— The introduction and assignments of programming activities can be embedded in children's familiar daily life experience, in order to engage them and connect especially the blind children to the idea of programming, computers, and robots. Including humor and prioritizing, initially, to have fun can also positively impacts children's attitude and involvement. Furthermore, it can be helpful in introducing abstract concepts to blind children.
— Providing elaborate direct instruction, tailored to the specific needs of certain children, can be worth giving some priority to and could work better than having children discover on their own with "in the moment support".

5.6 Limitations, Future Directions, and Conclusion

This study has some limitations. First, the small scope of the research, focusing on the experiences of a specific group of teachers, is in line with the aim at gaining an in-depth and explorative perspective of the teachers' own ideas and direct experience. This limits generalizability, the selection of materials included, and furthermore implies certain specific, technical information (such as with which version of materials teachers worked, or with which specific device or browser) is lacking. However, the wide, directly practical experience these teachers have to offer, both with children with visual impairments in general and with exploring programming with this group, can be seen as a vital element in gaining an understanding of how to optimize accessible programming education. The fact that some materials have not been explored by the teachers also points to the need to consider how best to keep the educational practice up to date on the continuously developing availability of programming materials for children. Second, a possible disadvantage of a focus group is that participants might not express their honest opinions. Although the influence of the set-up cannot fully be ruled out, the teachers appeared open and honest. This could be seen in the fact that they indicated when they had an idea or experience different from the others, and that they also indicated when they felt they could not answer something. Third, the transcript of the focus group was processed following the steps of the method of thematic analysis. There is some interpretation in this process of determining and labeling the excerpts. However, the steps of labeling all information and gathering the labels under themes [8, 10] were carefully followed and a second assessor was included in determining the correct labeling of the themes. Finally, although most issues that the teachers brought up directly concern the visual nature of materials or underlying concepts, it can be taken into account that certain materials also have challenges related to age or level and not specifically to vision. Specifically assembling robots and working on large and complex Lego projects [6] as well as the illogical working of the route of the Cubetto [14, 29] have been identified as difficult or problematic for children in general.

Several directions for future research can be pointed out. First, also when the aim is to identify materials suited for all children, it can be worthwhile to look into both possibilities and challenges for specific subgroups. This includes the potential of block-based programming for low vision children, and the specific challenges at the cognitive level for blind children. Concerning the latter, the complex area of understanding blind children's mental modeling and spatial navigation can be further explored in the context of specific programming assignments. Further, since unplugged programming has the potential as a highly suited inclusive tool, future studies could examine in detail the usability and accessibility of different unplugged materials for the group of low vision children as a whole. Second, the perspective of teachers in regular education would be an important addition. In this situation, where usually one blind or low vision child would work together with sighted peers, similar but also new questions arise on how to accommodate specific supports and facilitate collaborations. Related, assessment among sighted peers provides insights on their experience of the type of materials evaluated here, especially when working with a peer with a visual impairment and/or with a slightly adapted material. Finally, several of these directions as well as the general inquiry into inclusive programming materials can be addressed using structured observations of low vision, blind, and sighted children, and further, by connecting the use of materials to lessons, assignments, and instructions.

The perspective obtained through the experiences and ideas of teachers have worked in the practice of programming education with low vision and blind children, first of all, points toward concrete insights on the usability of the types of materials currently being used in education. Especially the value of unplugged lessons, but also the potential of work-arounds methods for inaccessible input and the suitedness of blockprogramming for low vision children are suggestions that can be directly made use of in educational practice. Furthermore, general considerations useful for the research and development of inclusive programming tools include the impact of fun and daily life connections, adding to an overall motivating and attention grasping experience, and careful explorations of aspects at the cognitive level. Second, both the ideas of the teachers and their tone/overall expression of searching highlight points of attention surrounding these concrete usability options. Most important here are the availability of and knowledge on different options, and the role of instructions and related possibilities of independent working when learning programming with different materials. Overall, adding to the knowledge that is growing on specific (often technical) issues with programming materials for low vision and blind children, further assessments could benefit from a comprehensive approach, taking into account dimensions such as how a material conveys programming concepts and the instruction surrounding the material.

ACKNOWLEDGMENTS

We would like to thank the teachers for sharing their experiences and ideas in the focus group.

REFERENCES

[1] Khaled Albusays and Stephanie 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.

[2] Khaled Albusays, Stephanie Ludi, and Matt Huenerfauth. 2017. 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*. Association for Computing Machinery, New York, NY, 91–100. DOI: https://doi.org/10.1145/3132525.3132550

[3] Hind Alotaibi, Hend S. Al-Khalifa, and Duaa AlSaeed. 2020. Teaching programming to students with vision impairment: Impact of tactile teaching strategies on student's achievements and perceptions. *Sustainability* 12, 13 (2020), 5320.

[4] Mihailo Antović, Austin Bennett, and Mark Turner. 2013. Running in circles or moving along lines: Conceptualization of musical elements in sighted and blind children. *Musicae Scientiae* 17, 2 (2013), 229–245.

[5] Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. Structjumper: A tool to help blind programmers navigate and understand the structure of code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3043–3052.

[6] Marina Umaschi Bers, Louise Flannery, Elizabeth R. Kazakoff, and Amanda Sullivan. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education* 72 (2014), 145–157. DOI: https://doi.org/10.1016/j.compedu.2013.10.020

[7] Jeffrey P. Bigham, Maxwell B. Aller, Jeremy T. Brudvik, Jessica O. Leung, Lindsay A. Yazzolino, and Richard E. Ladner. 2008. Inspiring blind high school students to pursue computer science with instant messaging chatbots. In *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*. 449–453. DOI: https://doi.org/10.1145/1352135.1352287

[8] Erik Blair. 2015. A reflexive exploration of two qualitative data coding techniques. *Journal of Methods and Measurement in the Social Sciences* 6, 1 (2015), 14–29.

[9] Stefania Bocconi, Silvia Dini, Lucia Ferlino, Cristina Martinoli, and Michela Ott. 2007. ICT educational tools and visually impaired students: Different answers to different accessibility needs. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*. 491–500. DOI: https://doi.org/10.1007/978-3-540-73283-9_55

[10] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.

[11] C. Cheong. 2010. Coding without sight: Teaching object-oriented java programming to a blind student. *8th Annual Hawaii International Conference on Education* (2010), 1–12. Retrieved from http://researchbank.rmit.edu.au/view/rmit:13231.

[12] Lynne Cooke. 2010. Assessing concurrent think-aloud protocol as a usability test method: A technical communication approach. *IEEE Transactions on Professional Communication* 53, 3 (2010), 202–215.

[13] Thomas J. Cortina. 2015. Reaching a broader population of students through "unplugged" activities. *Communications of the ACM* 58, 3 (2015), 25–27. DOI: https://doi.org/10.1145/2723671

[14] Hylke H. Faber, Josina I. Koning, Menno D. M. Wierdsma, Henderien W. Steenbeek, and Erik Barendsen. 2019. Observing abstraction in young children solving algorithmic tasks. In *Proceedings of the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. 95–106. DOI: https://doi.org/10.1007/978-3-030-33759-9_8

[15] Hylke H. Faber, Menno D. M. Wierdsma, Richard P. Doornbos, Jan S. van der Ven, and Kevin de Vette. 2017. Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network* 12, 1 (2017), 13–24. Retrieved from http://jeten-online.org/index.php/jeten/article/view/131.

[16] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. 2000. Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 345–352.

[17] Anita Gibbs. 1997. Focus groups. *Social Research Update* 19, 8 (1997), 1–8.

[18] Alex Hadwen-Bennett, Sue Sentance, and Cecily Morrison. 2018. Making programming accessible to learners with visual impairments: A literature review. *International Journal of Computer Science Education in Schools* 2, 2 (2018), 3. DOI: https://doi.org/10.21585/ijcses.v2i2.25

[19] Felienne Hermans and Ehimia Aivaloglou. 2017. A controlled experiment comparing plugged first and unplugged first programming lessons. *ACM International Conference Proceeding Series* (2017), 49–56. DOI: https://doi.org/10.1145/3137065.3137072

[20] Felienne Hermans and Marileen Smit. 2018. Explicit direct instruction in programming education. In *Proceedings of the PPIG*.

[21] Earl W. Huff Jr, Kwajo Boateng, Makayla Moster, Paige Rodeghero, and Julian Brinkley. 2021. Exploring the perspectives of teachers of the visually impaired regarding accessible k12 computing education. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 156–162.

[22] L'udmila Jašková and Mária Kaliaková. 2014. Programming microworlds for visually impaired pupils. *Proceedings of the 3rd International Constructionism Conference* (2014), 1–10. Retrieved from http://constructionism2014.ifs.tuwien.ac.at/papers/2.7_2-8251.pdf.

[23] Johan Jeuring, Gemma Corbalan, Joek˜van Montfort, Nienke˜van Es, and Hanneke Leeuwestein. 2016. Vorm en effect van Programmeeronderwijs in het primair onderwijs. *Technical Report Series* UU-CS-2016-005 (2016).

[24] Martina Kabátová, Ä¡udmila Jašková, Peter Lecký, and Vladimíra Laššáková. 2012. Robotic activities for visually impaired secondary school children. *3rd International Workshop, Teaching Robotics, Teaching with Robotics* (2012), 22–31.

[25] Ivan Kalas, Andrej Blaho, and Milan Moravcik. 2018. Exploring control in early computing education. In *Proceedings of the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.* Springer, 3–16.

[26] Aydın Kızılaslan, S. Levent Zorluoglu, and Mustafa Sözbilir. 2019. A hands-on classroom activity to teach science concepts for students with visual impairment. *Science Activities* 56, 4 (2019), 130–138.

[27] Varsha Koushik, Darren Guinness, and Shaun K. Kane. 2019. StoryBlocks: A tangible programming game to create accessible audio stories. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), 1–12. DOI: https://doi.org/10.1145/3290605.3300722

[28] Stephanie Ludi, Debra Bernstein, and Karen Mutch-Jones. 2018. Enhanced robotics! Improving building and programming learning experiences for students with visual impairments. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.* 372–377.

[29] Eva Marinus, Zoe Powell, Rosalind Thornton, Genevieve McArthur, and Stephen Crain. 2018. Unravelling the cognition of coding in 3-to-6-year olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In*Proceedings of the 2018 ACM Conference on International Computing Education Research.* 133–141. DOI: https://doi.org/10.1145/3230977.3230984

[30] Joanne McElligott and Lieselotte Van Leeuwen. 2004. Designing sound tools and toys for blind and visually impaired children. *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community.* 65–72. DOI: https://doi.org/10.1145/1017833.1017842

[31] Orni Meerbaum-Salant, Michal Armoni, and Mordechai (Moti) Ben-Ari. 2013. Learning computer science concepts with Scratch. *Computer Science Education* 23, 3 (2013), 239–264. DOI: https://doi.org/10.1080/08993408.2013.832022

[32] Lauren R. Milne and Richard E. Ladner. 2018. Blocks4All: Overcoming accessibility barriers to blocks programming for children with visual impairments. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–10. DOI:https://doi.org/10.1145/3173574.3173643

[33] Cecily Morrison, Nicolas Villar, Anja Thieme, Zahra Ashktorab, Eloise Taysom, Oscar Salandin, Daniel Cletheroe, Greg Saul, Alan F. Blackwell, Darren Edge, Martin Grayson, and Haiyan Zhang. 2018. Torino: A tangible programming language inclusive of children with visual disabilities. *Human-Computer Interaction* 00, 00 (2018), 1–49. DOI:https://doi.org/10.1080/07370024.2018.1512413

[34] Aboubakar Mountapmbeme and Stephanie Ludi. 2021. How teachers of the visually impaired compensate with the absence of accessible block-based languages. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility.* 1–10.

[35] Valerie Nesset and Andrew Large. 2004. Children in the information technology design process: A review of theories and their applications. *Library and Information Science Research* 26, 2 (2004), 140–161.

[36] Gary M. Olson and Judith S. Olson. 2003. Human-computer interaction: Psychological aspects of the human use of computing. *Annual Review of Psychology* 54, 1 (2003), 491–516.

[37] Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri. 2017. Reviewing the affordances of tangible programming languages: Implications for design and practice. *IEEE Global Engineering Education Conference* 1811–1816. DOI: https://doi.org/10.1109/EDUCON.2017.7943096

[38] Ana Cristina Pires, Filipa Rocha, Antonio José de Barros Neto, Hugo Simão, Hugo Nicolau, and Tiago Guerreiro. 2020. Exploring accessible programming with educators and visually impaired children. In *Proceedings of the Interaction Design and Children Conference.* 148–160.

[39] Thomas W. Price and Tiffany Barnes. 2015. Comparing textual and block interfaces in a novice programming environment. In *Proceedings of the 11th Annual International Conference on International Computing Education Research.* Association for Computing Machinery, New York, NY, 91–99. DOI:https://doi.org/10.1145/2787622.2787712

[40] Alexandra Queirós, Anabela Silva, Joaquim Alvarelhão, Nelson Pacheco Rocha, and António Teixeira. 2015. Usability, accessibility and ambient-assisted living: A systematic literature review. *Universal Access in the Information Society* 14, 1 (2015), 57–66. DOI:https://doi.org/10.1007/s10209-013-0328-x

[41] Janet C. Read, Stuart MacFarlane, and Chris Casey. 2002. Endurability, engagement and expectations: Measuring children's fun. In *Proceedings of the Interaction Design and Children.* Shaker Publishing Eindhoven, 1–23.

[42] Paul J. Riccomini, Stephanie Morano, and Charles A. Hughes. 2017. Big ideas in special education: Specially designed instruction, high-leverage practices, explicit instruction, and intensive instruction. *Teaching Exceptional Children* 50, 1 (2017), 20–27.

[43] Wouter J. Rijke, Lars Bollen, Tessa H. S. Eysink, and Jos L. J. Tolboom. 2018. Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education* 17, 1 (2018), 77–92.

[44] Yvonne Rogers, Venus Shum, Nicolai Marquardt, Susan Lechelt, Rose Johnson, Howard Baker, and Matt Davies. 2017. From the BBC micro to micro: Bit and beyond: A British innovation. *Interactions* 24, 2 (2017), 74–77. DOI: https://doi.org/10.1145/3029601

[45] Jaime Sánchez and Fernando Aguayo. 2005. Blind learners programming through audio. In *Proceedings of the CHI'05 Extended Abstracts on Human Factors in Computing Systems.* 1769–1772.

[46] Ather Sharif, L. McCall Aedan, and R. Bolante Kianna. 2022. Should i say "disabled people" or "people with disabilities"? Language preferences of disabled people between identity-and person-first language. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility.* Association for Computing Machinery, New York, NY, To Appear.

[47] Janet Smithson. 2000. Using and analysing focus groups: Limitations and possibilities. *International Journal of Social Research Methodology* 3, 2 (2000), 103–119.

[48] Andreas Stefik, Richard E. Ladner, William Allee, and Sean Mealin. 2019. Computer science principles for teachers of blind and visually impaired students. *ACM Inroads* 10, 2 (2019), 50–57. DOI: https://doi.org/10.1145/3324894

[49] Andreas M. Stefik, Christopher Hundhausen, and Derrick 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.

[50] David W. Stewart, Prem N. Shamdasani, and Dennis W. Rook. 2007. Analyzing focus group data. *Focus Groups: Theory and Practice* 20 (2007), 115–140.

[51] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. 2016. How people with low vision access computing devices: Understanding challenges and opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility.* 171–180.

[52] Catherine Thinus-Blanc and Florence Gaunet. 1997. Representation of space in blind persons: Vision as a spatial sense? *Psychological Bulletin* 121, 1 (1997), 20.

[53] Gabriella Tisza and Panos Markopoulos. 2021. Understanding the role of fun in learning to code. *International Journal of Child-Computer Interaction* 28 (2021), 100270.

[54] Tomaso Vecchi, Carla Tinti, and Cesare Cornoldi. 2004. Spatial memory and integration processes in congenital blindness. *Neuroreport* 15, 18 (2004), 2787–2790.

[55] Benjamin Wohl, Barry Porter, and Sarah Clinch. 2015. Teaching computer science to 5–7 year-Olds: An initial study with scratch, Cubelets and unplugged computing. *Proceedings of the Workshop in Primary and Secondary Computing Education.* 55–60. DOI: https://doi.org/10.1145/2818314:2818340

[56] Junnan Yu and Ricarose Roque. 2018. A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children* (2018), 289–299. DOI: https://doi.org/10.1145/3202185.3202738

Received 30 June 2020; revised 24 December 2021; accepted 19 August 2022