An exploration of unplugged programming education for elementary school children who have low vision or are blind

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

An Exploration of Unplugged Programming Education for Elementary School Children Who Have Low Vision or are Blind

Anna van der Meule, Mijke Hartendorpb, Wendy Voornc and Felienne Hermansad

Abstract

Although programming education provides a valuable introduction to computers, accessible materials for young learners with vision impairments are lacking. This study focused on unplugged programming, using the sandwich robot lesson, for young learners who have low vision or are blind. 17 children with vision impairments participated in pairs in a programming assignment where one child instructed (‘’programmed’’) the other child to prepare a sandwich. The findings, based on thematic coding of children’s behaviors, show the children could access the artifacts visually or tactile, worked with enthusiasm and relatively independent, and could complete the assignment. The children who are blind did require additional assistance, and finding correct instructions was challenging. Our insights suggest the promise of the unplugged sandwich robot lesson as an inclusive programming tool.

Introduction

Familiarity with digital and computational thinking skills, part of the “21st century skills”, is vital to navigate the current world, both in order to understand daily life and to apply digital skills in professional lives. Consequently, it is essential to ensure that all young learners, including individuals with impairments, are being introduced to these skills (Prado, Jacob, & Warschauer, 2021). Specifically in the context of early programming education, one fundamental element of an inclusive approach concerns usable and suited materials. A wide range of programming materials for young learners exists and continues to be developed, yet adequate insights in usable materials for especially younger learners with vision impairments have until recently remained lacking (Morrison et al., 2018). Challenges are known concerning the often-visual nature of programming materials for children (Milne & Ladner, 2018; Morrison et al., 2018), including the widely used block-based programming languages and tangible floor robots or robotic sets (Hadwen-Bennett, Sentance, & Morrison, 2018; Jašková, & Kaliaková, 2014, Kabátová et al., 2012). These materials are consequently less or not accessible for children who are blind or have low vision, which contributes to lack of full participation in programming lessons. One option to improve this lies in adapted or newly constructed versions of these materials (Milne & Ladner, 2018; Morrison et al., 2018).

Another valuable but less explored option could be found in the form of unplugged programming tools. These tools are characterized by the absence of the use of computer or electronics, instead they rely upon usually few, easily adaptable daily artifacts (Cortina, 2015), for example beads, cups or cards (Bell, Alexander, Freeman, & Grimley, 2009; Faber, Wierdsma, Doornbos, van der Ven, & de Vette, 2017; Hermans & Aivaloglou, 2017). Unplugged lessons using these tools can be highly engaging (Wohl, Porter, & Clinch, 2015), and have been shown to contribute to the understanding of computer science concepts (Hermans & Aivaloglou, 2017). Moreover, teachers of learners with vision impairments recently expressed their enthusiasm about unplugged programming as the most inclusive tool currently available for their learners (van der Meulen et al., 2022).

Unplugged materials have the potential for children who are blind or have low vision to equally participate with their peers in programming lessons. This study consists of a qualitative exploration into unplugged programming for elementary school learners with vision impairments. The specific unplugged activity of the sandwich robot is used, where children program each other to prepare a sandwich. The rationale behind this selection is that this is a typical unplugged tool (using daily artifacts that are actively engaged with) that is currently popular in the Dutch school context. In our exploration of this tool, our objectives, based on usability and accessibility criteria (Queirós et al., 2015), are to gain insight in: 1) through which sense modalities (vision, touch, auditory) access to the material is obtained, and whether this can be done independently, 2) how learning and working on the assignment takes place, 3) the occurrence of collaboration, creativity, and positive and negative experience. Our aim is to assess how learners with vision impairments approach and experience an unplugged programming activity.

Unplugged programming tools

Unplugged or offline programming, as compared to plugged, refers to learning how to program without a computer or any types of electronics (Bell et al., 2009; Faber et al., 2017). Unplugged materials form one type of materials currently available to teach children programming, examples of other types include floor robots or mini-computers, child-level programming languages, or games (Yu & Roque, 2018).

Unplugged lessons use artifacts such as beads or cups, or have the participants themselves act out an activity (Bell et al., 2009; Faber et al., 2017; Hermans & Aivaloglou, 2017). A design principle is that the activities are easy to implement, fun and engaging, and invite active, collaborative behaviour (Bell et al., 2009; Cortina, 2015). Children for instance learn about binary counting by deciphering and creating binary code using paper cut outs that represent binary numbers, or learn about algorithms by taking on the role of programmer to write an algorithm to complete a specific action (Faber et al., 2017). Research on unplugged programming, though not that extensive yet, highlights the enthusiasm of teachers and students (Brackmann et al., 2017; Faber et al., 2017), and indicates its effectiveness in getting children engaged in (Wohl et al., 2015) and mastering (Hermans & Aivaloglou, 2017) computer science concepts.

An additional essential feature of unplugged programming tools is its potential to reach a broad population (Cortina, 2015). This includes learners with limited access to computers or internet, but also learners with impairments (Cortina, 2015; Faber et al., 2017). The activities generally require few artifacts, easily adapted for groups with different possibilities and needs (Cortina, 2015). This can be especially relevant for learners with vision impairments, since this group is diverse and challenging in their use of and preferences in technology. The often very visual nature of plugged programming materials for children can result in issues for children who are blind or have low vision (Hadwen-Bennett et al., 2018; Jašková, & Kaliaková, 2014; Kabátová et al., 2012; Morrison et al., 2018). Depending on the specific material, difficulties have been identified in incompatibility with assistive technologies such as screenreader technology, output in the form of animations (Hadwen-Bennett et al., 2018, Morrison et al., 2018, and in the case of tangible tools in the form of visual properties or small pieces (Kabátová et al., 2012; Morrison et al., 2018).

Some potential drawbacks can also be mentioned to this overall promising picture of unplugged programming. Compared to working with computers or robots, unplugged programming can appear less exciting and be less inviting (Faber et al., 2017; Wohl et al., 2015). Furthermore, when the connection between unplugged activities and plugged activities or computer science is not adequately understood, exposure to unplugged programming can lessen the enthusiasm for computer science careers (Taub, Armoni, & Ben-Ari, 2012).

Unplugged programming for children with vision impairments

Interestingly, there seems to be very little exploration so far in the potential of unplugged tools for learners with vision impairments. A recent literature review on accessibility of programming tools for learners with vision impairments does not mention unplugged tools (Hadwen-Bennett et al., 2018). Further, several studies look into new or adapted tangible electronic tools or alternative programming environments (Jašková, & Kaliaková, 2014; Kabátová et al., 2012; Koushik, Guinness, & Kane, 2019; Milne & Ladner, 2018; Morrison et al., 2018). Reasons for the lack of attention for unplugged might include that most research into programming for this group has focused on older learners, for whom unplugged can seem (though not necessarily is) less suited, that the focus often lies in the identification of electronic issues, and that certain plugged tools are currently very popular. In a recent focus group study however, teachers of learners with vision impairments at the elementary school level were within their comparison of different programming materials especially enthusiastic about unplugged tools (van der Meulen et al., 2022). They experienced unplugged lessons as the only approach that can engage low vision and blind learners equally. Another exploration of unplugged tools for high school learners with vision impairments highlights its value but indicates potential issues with visual artifacts and metaphors (Stefik et al., 2019).

Our structured exploration of unplugged programming for learners with vision impairments is grounded in the concepts of usability (referring to effectiveness, efficiency and experience in the use of a product) and accessibility (the extent to which everyone can use the product) (Queirós et al., 2015). An essential concept in the user experience of children is “fun”, which captures the adult equivalence of user satisfaction (Read, MacFarlane, & Casey, 2002). Previous usability research on programming materials, for sighted children and children with vision impairments, has focused on whether and how a material could be accessed, level of support and instruction that was needed, possibility for children to collaborate and be inventive, and the valence of the overall experience (Donker & Reitsma, 2004; Milne & Ladner, 2018; Morrison et al., 2018; Read et al., 2002; Van Kesteren et al., 2003). Accessibility should in this context include the diversity within vision impairments (with most having a diverse range of low vision, and a smaller group being blind).

We have selected the sandwich robot lesson as an unplugged tool to explore in learners with vision impairments. In this lesson, children are required to program the teacher or each other as a robot to prepare a sandwich (Bagge). By composing a set of specific step by step instructions for “the robot”, students learn about the basic working of a robot and the instruction of an algorithm (Faber et al., 2017). The sandwich robot is a typical example of an activating, engaging unplugged lesson, that moreover uses familiar daily artifacts. In our inquiry of this tool, we distinguish three main objectives, building upon previous insights into usability and accessibility in the context of programming materials for learners with vision impairments. We aim to observe 1) through which sense modalities (vision, touch, auditory) access to the material is obtained, and to what extent can this be done independently, 2) how learning and working on the assignment takes place, 3) the occurrence of collaboration, creativity, and positive and negative experience. We expect to provide insights into how learners who are blind or have low vision approach and experience unplugged programming, which will give an indication of the potential of such tools for this group of learners.

Method

Participants

Seventeen children (11 boys and 6 girls) participated, divided over eight pairs (one of which contained three children). The children came from three classes from different special education schools, which are part of the two Dutch expertise centres for individuals with vision impairments and are located in different parts of the Netherlands. Two classes were what is indicated as the “upper level” of elementary school (10-12 years old) and one the “lower level” (6-8 years old). One of the upper level as well as the lower level class had some previous experience with programming. Twelve of the 17 children had low vision, the five other children were indicated as “braille students”, three of them had very limited residual sight and two were completely blind. Because in the Netherlands children with vision impairments are enrolled in special education only if they cannot partake in regular education, the children had additional special learning needs.

Procedure

Schools and classes were recruited through the two expertise centres for individuals with vision impairments. In the context of a larger project on programming materials for children with vision impairments, all participating classes received three programming lessons in which a different material was introduced. The unplugged lesson was used as the first lessons in the three classes included in this study. The project was approved by the ethical committee of the faculty of Science at the University of Leiden. Parents were informed through the teachers with informed consent letters, in which it was explained that three programming lessons would be given in the class of the children, during which all children would be present. Parents were asked to give permission for the children to participate in the research and for video recording during this lesson. If parents did not give permission, children would still be able to take part in the lesson but no video recording would be made and no data would be collected. In the three groups participating in the unplugged lesson, half to all of the parents gave consent. Children who did not have consent worked in a separate classroom in order to ensure they would not appear in the recordings. During the assignment each pair of children was guided and supported by either the main researcher or a research-assistant (both indicated below as the tester). The research-assistants were students in social sciences or computer science bachelor tracks. They received training beforehand on working with children with vision impairments (including how to provide instructions and support) as well as on facilitating and monitoring the set-up of the assignment and the constructive interaction (explained below).

The sandwich robot assignment and test setup

The unplugged lesson and assignment of the sandwich robot was based on the Dutch teaching material describing the “chocolate sprinkles robot”, known also as “jam sandwich algorithm” (Bagge). The lesson started with a brief plenary introduction in which the topic of writing an algorithm in the form of a step-by- step instruction and taking into account a robot’s inability to think for himself, was explained. The children were divided up into pairs and matched to a tester. After the instruction on the protocol (explained below), the tester started the camera, and explained that the children would program each other to prepare a sandwich with chocolate sprinkles. After the roles of robot and programmer were divided (following the preference of the children), specific instructions for both roles were provided, indicating to the programmer that they had to come up with and give specific instructions to the robot, and to the robot that they had to follow the instruction by the programmer and not think themselves. The tester placed the materials in front of the children (plate, knife, butter, chocolate sprinkles, bag with two sandwiches) and the children could get started.

Table 1 summarizes the eight pairs of children. The children within a pair are referred to by their roles in the assignment: “robot-child” and “programmer-child”. The protocol of constructive interaction method was followed during the assignment (Als et al., 2005). Within a collaborative setting between children, they are stimulated to verbalize their thoughts in order to optimize insight in their experiences. An elaborate instruction on thinking aloud was provided by the tester at the start of the session, including an example and practice (Als et al., 2005; Donker & Reitsma, 2004; Van Kesteren et al., 2003). During the session itself the children the tester used neutral prompts to remind the children to verbalize and to work together. Further, recommendations for usability testing with learners with impairments were followed by working in individually guided pairs, and using tailored support (Foss et al., 2013; Guha, Druin, & Fails, 2008). The teachers helped form pairs of children who worked well together and if needed advised on individual children.

Table 1: Characteristics of child pairs

Pair	**Robot-child**	**Programmer-child(ren)**	**School level**
1	Braille (f)	Braille (m)	Lower
2	Braille (m)	Low vision (m)	Lower
3	Braille (m)	Braille (m)	Higher
4	Low vision (m)	Braille (m)	Higher
5	Low vision (f)	Low vision (f)	Higher
6	Low vision (m)	Low vision (m)	Higher
7	Low vision (f)	Low vision (m)	Higher
8	Low vision (m)	Low vision (f) (both)	Higher

Data processing

All sessions were individually recorded on video. A combination of coding and transcribing was used to process non-verbal and verbal data, using a detailed pre-defined coding scheme. This approach fits a thematic analysis of qualitative data (Braun & Clarke, 2006). The scheme included 17 categories of behaviour, 9 of which were used for the current research and are described in Table 2. Other categories referred for instance to behaviour relevant only for materials with electronics. Each category contained several pre-specified, directly observable behaviours (Donker & Reitsma, 2004) and the option to indicate a non-pre-specified behaviour fitting the category. The categories and pre-specified behaviors were derived from usability and accessibility concepts and previous observational research on programming tools by sighted children and children with vision impairments (Donker & Reitsma, 2004; Faber et al., 2019; Milne & Ladner, 2018; Morrison et al., 2018; Read et al., 2002; Van Kesteren et al., 2003) and if necessary adapted or complemented to fit the current purpose. As indicated in Table 2, the categories “independent access” (referring to whether and through which sense modalities the material could be independently used) and “access through other” provide insight into the first objective of the study; the categories “learning material” and “working on assignment” connect to the second objective, and the remaining five categories on collaboration, creativity and overall experience connect to the third. Further, verbal behaviours were transcribed verbatim for four pairs (pairs 1, 2, 3 and 6), after which saturation was reached. Consequently, for the remaining four pairs behaviours were coded but not transcribed. The eight completed coding schemes (one for each pair) were further processed for each category by collecting information per category across the pairs. The information from the transcriptions was used to clarify and deepen understanding of the children’s behaviour (for instance, assessing the content of positive verbalisation’s). Finally, the categories were gathered per study objective, and summarized.

Table 2: Study objectives, behavioural categories and sample behaviours of coding scheme

Study objective	**Behavioral category**	**Sample behaviors**
Independent access, sense modality, access through other	Independent access	Follow visually, follow tactile, follow auditory, material to child, child to material
Independent access, sense modality, access through other	Access through other	Ask explanation child, ask explanation tester, receive explanation child, receive explanation tester, tactile guidance other child, tactile guidance tester
Learning material, working on assignment	Learning Material	Explore visually, explore tactile, explore auditory, explore by actions, explore by reasoning, listening to instruction, experience material as logical, experience material as not logical, ask questions during instruction, ask questions during assignment, struggling to learn, learning easily

Study objective	Behavioral category	Sample behaviors
Learning material, working on assignment	Working on Assignment	Starting, finishing, working on assignment, playing or joking with material, trying material, working outside of material, not doing anything
Collaboration, creativity	Presence collaboration	Non-verbal contact, taking in roles, explaining to include, explaining to discuss
Collaboration, creativity and exploration, experience	Absence Collaboration	Work opposite goals, competing, work different outputs, one child doing something else
Collaboration, creativity and exploration, experience	Creativity & exploration	New step with used element, new step with new element, initiate new assignment, conduct new idea assignment
Collaboration, creativity and exploration, experience	Positive experience	Laughing, smiling, moving excitedly, positive verbalisation, sitting/lying comfortably
Collaboration, creativity and exploration, experience	Negative experience	Frowning, showing boredom, negative verbalisation, shrugging, showing distraction, showing confusion or difficulty

Results

Independent access, sense modality, and access through other senses

In five of the eight pairs both children accessed the material entirely visually. Further, in Pair 3 both children accessed the material tactile, in Pair 2 one child had visual access while the other had tactile access, and in Pair 1 one child accessed the material entirely visual whereas the other child combined visual and tactile access. When accessing visually, incidentally the child brought the material closer to their face, and more often the child brought their face closer to the material. Two children who had tactile access also brought themselves or the material closer a few times. Finally, not all children were clear about their possibilities. The children of Pair 1 (both braille students with residual vision) used a not transparent combination of a visual and tactile approach. Other children however were very outspoken about their needs and preferences in order to follow the scene. In Pair 3 for instance, the child who was blind and who played the programmer immediately took initiative to stand behind and put his hands on the shoulders of the robot-child to follow his movement: Tester: “Where would you start?” Programmer-child: “Can I just stand behind him...if he has to make a movement he should.”(Programmer-child stands behind the robot-child). Getting access to the material through the other child or tester was only observed for the children who are blind (occasionally in Pair 1, and frequently in Pair 2 and 3). Most frequent was the child who was blind receiving an explanation from the tester, which often involved the tester directly describing the visual scene. As Pair 2 showed:

> Tester: “Maybe then the robot gets confused, since there are two sandwiches in the bag.” Programmer- child: “Take one sandwich out of the bag”. Robot-child: “This is one sandwich.” Tester: “Go ahead and feel, there are two”.

Other behaviours in Pair 3 included the programmer-child asking and receiving an explanation from the tester, and (once) the programmer-child receiving an explanation from the other child. In Pair 2, the robot-child received explanations from the other child a few times, as well as tactile guidance by the tester or the programmer-child. Quite often this involved combinations of behaviours, for instance receiving an explanation from both the tester and the other child as well as tactile guidance by the other child: Programmer-child (takes the hand of the robot-child with the knife and adjusts the position):

> “This is how you do that, robot”. Tester: “No it is actually a little stuck in the butter. Help a bit, programmer.” Programmer-child (takes the hand of the robot-child and pulls the knife out of the butter). Tester: “The knife got a little stuck”. Programmer-child: “Yes”. Robot-child: “Is it done?” Programmer- child: “Yes.”

Finally, in this pair it was also observed once that the programmer-child handed the robot-child material, and occasionally the low vision programmer-child received explanations from the tester.

Learning the material and working on the assignment

Learning how to work with the material most often, frequently in all pairs, involved listening to instructions by the tester. After the initial instruction, the tester intervened by asking for clarification, indicating an incorrect instruction, or providing suggestions. Pair 6 for example showed:

> Tester: “But ’take’ is actually a pretty vague instruction. What exactly should the robot do to take the butter?”

Most pairs (except Pair 8) occasionally also showed other learning behaviours, primarily asking questions and struggling to learn to work with the material. Struggling could be seen for both the robot-child and programmer-child, for example in Pair 2:

> Tester: “A lot but not too much, would he understand that?” Robot-child: “Yes.” Programmer-child: “He understands that.”

Other behaviours were seen incidentally: exploring by reasoning, easily learning the material and experiencing material as logical as well as experiencing material as not logical. Some “other” coded behaviours were the tester instructing the children who don’t respond, or the child giving an explanation to the tester.

All pairs started the assignment after receiving instruction, and managed to finish the assignment, spending between 9 (Pairs 1 and 8) and 25 minutes (Pair 4). Different types of instructions could be seen. First of all, instructions in the form of specific small and distinguishable actions were seen in all four pairs who’s verbal behaviour was transcribed (most often in Pair 2 and 3, occasionally in Pair 1 and 6). Examples of such instructions are: “Right hand out”; “Turn the knife a little more”; “Let go of the knife”; “Walk forward”. Pair 3 almost exclusively showed these types of instructions, within a long sequence of steps. Second, very short instructions following or complementing a previous instruction occurred occasionally in Pairs 6 and 2, and often in Pair 3: “Stop”; “Enough”. Third, a more broad type of instruction was most often seen in Pairs 1 and 6. The programmer-children of these pairs continued with these instructions even after multiple attempts of the tester to break this up into smaller actions: “Put the sandwich on the place”; “Get the chocolate sprinkles”. Finally, specifically in Pair 2 the low vision programmer-child was searching for clear instructions for the blind robot-child:

> Programmer-child: “Feel where the sandwich is and then you know where the sandwich is and then there you can put on chocolate sprinkles”.

All pairs also joked and played with the materials. For some pairs this occurred occasionally at the beginning or end. Pair 2 and 6 continued joking through the assignment, disrupting the assignment for example by throwing the sandwich on the floor. Other behaviours outside the assignment yet with the material involved reflecting on or discussing the material (often but not always at the end) or comments on or playing with the food. A few times children displayed unrelated behaviour, for instance distraction by other children.

Collaboration, creativity and exploration, experience

Collaborative behaviours were seen in each pair, yet with the exception of Pair 2 only occasionally. Often the programmer-child and the robot-child corrected each other. For instance, the robot commented on incorrect instructions.

> Pair 1: Programmer-child: “Put a lot of chocolate sprinkles on the sandwich but not too much.” Robot-child: “Can’t.” Tester: “What is it that can’t?” Robot-child: “I don’t understand.”

Further, role shifting occurred relatively often, most often by the robot- child coming out of the robot role:

> Pair 2:) Tester: “Maybe you should tell what is happening. Do we have enough butter?” Programmer-child: “No.” Robot-child (with robot voice): “I’ll get some more.”

Other incidental behaviours were asking for advice and explaining in order to include (both only in Pair 2), taking in roles and several “other” behaviours including giving advice or help or discussing the situation.

Correcting, advising, and shifting in roles all occasionally occurred together.

Creative or exploring behaviours were seen incidentally (in Pair 6 and 7) or often (in Pairs 2 and 3). Generally, this was initiated by the robot-child, who spontaneously gave shape to this role by using a robot voice or sounds (both coded as “other” behaviour). All four pairs displayed this, the robot in Pair 2 even consistently through the entire assignment:

> Robot-child (with robot voice): “Can I start?” Tester: “The robot is completely in his robot mode.” Robot-child: “Bzzzt”.

In Pair 3, several times the robot-child added an unexpected issue that the programmer had to deal with, such as being low on battery or initiating an automatic cleaning process:

> Robot-child (with robot voice): “Battery 5 percent” (tester and programmer-child laugh). Programmer-child (goes to the robot and puts his hands on his shoulders). Tester: “Well we have to...” Robot-child: “Powering down in 5-4-...”.

Twice a pre-defined behaviour occurred: once in the form of a new step with a used element and once a new step with a new element (eating the sandwich or bringing the knife to the kitchen).

Finally, all pairs showed positive behaviours, including all pre-defined behaviours. By far the most common behaviour was laughter, which was seen in all pairs. This occurred a few times (Pair 1) to very often (Pair 6 and 8). Positive verbalisations also occurred often (in six pairs), primarily by expressing enthusiasm at the end of the assignment or while evaluating the material. In Pair 2 the robot-child expressed positive comments through the assignment: Robot-child: “I like it, nicely playing robot”. Further, smiling was seen in four pairs and sitting/lying comfortably as well as moving excitedly was only seen (multiple times) in pair 2 (where the robot was often humming contently). “Other” behaviours included being enthusiastic to start or disappointed to stop or showing surprise. Negative experiences were also shown in all pairs, with confusion or experience of difficulty occurring most often (except in Pair 3). Mostly this behaviour occurred a couple times, in Pairs 2 and 5 quite frequently. Almost always it was the robot-child who experienced difficulty, related to them noticing an incorrect instruction and being uncertain how to proceed since the robot was not allowed to think.

Pair 5: Robot-child (stops putting on butter by himself): “Hmmm so difficult.” Tester: “Hmmm:”. Robot-child: “...can’t think”.

Occasionally the children showed distraction, frowned, expressed themselves negatively or got upset. “Other” negative experiences included showing frustration, looking questioningly at the tester, or showing doubt about the other child. The robot-child of pair 5 was very critical overall, and the programmer-child of Pair 7 was struggling a lot.

Discussion

This study consisted of a structured qualitative exploration into unplugged programming, using the sandwich robot tool, for young learners with vision impairments. Seventeen children who are blind or have low vision were observed while they worked in pairs on the assignment. Below, the insights are discussed for the three objectives of this study.

Sense modalities, independent access and assistance while learning and working

All children were able to complete the assignment. Importantly, the children appeared to be able to follow their preferred sense modality to gain access, which was primarily visually and tactile. The same activity, for instance keeping track of the robot-child carrying out an instruction, was followed visually by the children who have low vision and tactile by the children who are blind. The possibility for this “hybrid approach” (Morrison et al., 2018) is a very valuable feature for children with vision impairments, fitting the diversity of their possibilities and preferences (Bocconi, Dini, Ferlino, Martinoli, & Ott, 2007; Milne & Ladner, 2018; Morrison et al., 2018). Although all children could consequently fully participate, it was apparent that the children who are blind did require additional support either by the tester or the other child. This type of verbal or tactile assistance was, to different extents, needed by all children who are blind, and not by the children who have low vision. The required guidance to capture the visual scene is similar to the previously found issues in unplugged programming with visual artifacts (Stefik et al., 2019).

Children who are blind appear not able to participate completely equally in the sandwich robot lesson. However, the extent and form of the required support (a brief verbal hint or manual readjusting of a hand) seems relatively minor, compared to adaptations necessary for children who are blind to work with plugged tools (Hadwen-Bennett et al., 2018). This confirms the adaptive and flexible nature of unplugged programming (Cortina, 2015) and its especially high suitedness for learners with vision impairments (van der Meulen et al., 2022). However, it also stresses the need to provide formal guidelines for required additional support when children who are blind work with unplugged tools (Van Mieghem, Verschueren, Petry, & Struyf, 2020).

Collaboration, creativity, and positive and negative experience

The children displayed a positive experience of the tool and assignment, enjoying themselves and working concentrated and at ease. Negative experiences such as frustration or distraction occurred only incidentally. From the perspective of usability assessments, this confirms a high user satisfaction (Read et al., 2002), in line with unplugged tools being conceived of as fun and engaging (Cortina, 2015, van der Meulen et al., 2022). Further, unplugged programming has elsewhere been indicated as less exciting and novel compared to computers or robots, which can however be seen both as a disadvantage (Faber et al., 2017; Wohl et al., 2015) or as adding to its suitedness as a calm and natural start of programming education, without the distraction and added cognitive load of the novelty of computers (Hermans & Aivaloglou, 2017). The latter interpretation might apply especially to groups of learners with impairments, as can be seen in the enthusiastic but also concentrated overall attitude of our group.

Further, although creativity is limited in the defined assignment of the sandwich robot, the children displayed surprising spontaneous actions where they played the robot and added their own ideas. In addition to being inventive and fun, such additions (such as the robot being low in battery) also display an accurate expression of unpredictable, interfering aspects of robots or computers that have to be dealt with instantly.

Most children also experienced some difficulty or confusion, especially when incorrect instructions were given. Most relevant for the usability of the tool for the target group was the example where a child with low vision struggled specifically to form understandable instructions for a child who was blind. It should be further explored how children with different visions, including sighted children, collaborate with this tool. The majority of children with vision impairments take place in mainstream education, which stressed the need to assess how collaborations between differently sighted children are experienced on both sides. A related potential issue with the instructions in the sandwich robot tool is that the child playing the robot can him or herself determine whether an instruction is correct. Consequently, it needs to be assessed how instruction from the teacher can be optimized here, as well as the effectiveness of teaching programming through the sandwich robot (as has been explored for other unplugged approaches (Hermans & Aivaloglou, 2017 and Wohl et al., 2015)).

Limitations, future directions, and conclusion

A first limitation of this study concerns the focus on materials, and not other aspects, of programming education. To provide sound and inclusive programming education it is also necessary to formally determine teaching instructions and additional teacher or peer support for specific groups (Van Mieghem et al., 2020). Second, it is important to further consider the potential of unplugged programming for older learners with vision impairments. The current focus was on young learners also because programming materials for children are often especially visual (Morrison et al., 2018), yet difficulties persist for older learners as well. A recent workshop with young adults with vision impairments points towards the value for them as well to first explore algorithmic thinking and problem solving before getting started with the plugged programming environments (Alotaibi, S Al-Khalifa, & AlSaeed, 2020). Third, the approach in the data collection of balancing between a fixed protocol and providing flexibility to facilitate individual needs in the target group can be considered the valid approach in usability research with children with impairments (Foss et al., 2013; Guha et al., 2008). However, the resulting differences between the sessions in terms of length and support by the tester could have impacted for instance the impression of children's independence and collaboration. In addition, thinking aloud was actively stimulated but, similar to previous studies (Als et al., 2005; Donker & Reitsma, 2004), some children appeared notably more verbal than others which possibly conceals experiences which are difficult or negative. Finally, concerning the data processing, an approach was chosen to optimize extensive and detailed depictions of the children's sessions fitting the purpose of this qualitative study (Tracy, 2010). A limitation can be seen in the reliance upon one coder, however, accuracy was strived for by using a fixed and precise coding scheme.

Overall, the sandwich robot appears an engaging and activating activity which learners who are blind or have low vision can participate in through their preferred sense modality, requiring for the learners who are blind some additional support. As a typical unplugged tool, it is flexible to use and suited for a group with different, and divers, possibilities and needs. The sandwich robot, as well as potentially other unplugged tools, can be a valuable inclusive programming education tool, providing children who are blind or have low vision the possibility to be introduced to digital and computational thinking skills, together with their peers.

Acknowledgments

We would like to thank all the children who participated in the studies and the schools who facilitated their participation, as well as the students who helped during the data collection.

References

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

Als, B. S., Jensen, J. J., & Skov, M. B. (2005). Exploring verbalization and collaboration of constructive interaction with children. *Ifip conference on human- computer interaction*, 443–456.

Bagge, P. Retrieved July 2022, from http://code-it.co.uk/new-jam-sandwich-algorithm/

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20–29.

Bocconi, S., Dini, S., Ferlino, L., Martinoli, C., & Ott, M. (2007). ICT educational tools and visually impaired students: Different answers to different accessibility needs. *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4556 LNCS(PART 3), 491–500.

Brackmann, C. P., Roman-Gonzalez, M., Robles, G., Moreno-Le´on, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th workshop on primary and secondary computing education,* 65–72.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology, 3*(2), 77–101.

Cortina, T. J. (2015, February). Reaching a broader population of students through “unplugged” activities. *Commun. ACM*, 58 (3), 25–27.

Donker, A., & Reitsma, P. (2004). Usability testing with young children. *Proceedings of the 2004 conference on interaction design and children: building a community*, 43–48.

Faber, H. H., Koning, J. I., Wierdsma, M. D., Steenbeek, H. W., & Barendsen, E. (2019). Observing abstraction in young children solving algorithmic tasks. *International conference on informatics in schools: Situation, evolution, and perspectives*, 95–106.

Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, 12, 13–24.

Foss, E., Guha, M. L., Papadatos, P., Clegg, T., Yip, J., & Walsh, G. (2013). Cooperative inquiry extended: Creating technology with middle school students with learning differences. *Journal of Special Education Technology, 28*(3), 33–46.

Guha, M. L., Druin, A., & Fails, J. A. (2008). Designing with and for children with special needs: an inclusionary model. *Proceedings of the 7th international conference on interaction design and children*, 61–64.

Hadwen-Bennett, A., Sentance, S., & Morrison, C. (2018). Making Programming Accessible to Learners with Visual Impairments: A Literature Review. *International Journal of Computer Science Education in Schools, 2*(2), 3.

Hermans, F., & Aivaloglou, E. (2017). A controlled experiment comparing plugged first and unplugged first programming lessons. *ACM International Conference Proceeding Series*, 49–56.

Jašková, L., & Kaliaková, M. (2014). Programming microworlds for visually impaired pupils. *Proceedings of the 3rd international constructionism conference*.

Kabátová, M., Jašková, L., Lecký, P., & Laššáková, V. (2012, April). Robotic activities for visually impaired secondary school children. *3rd international workshop, teaching robotics, teaching with robotics*, 22-31.

Koushik, V., Guinness, D., & Kane, S. K. (2019). StoryBlocks: A tangible programming game to create accessible audio stories. *Conference on Human Factors in Computing Systems - Proceedings*, 1–12.

Meulen, A. V. D., Hartendorp, M., Voorn, W., & Hermans, F. (2022). The perception of teachers on usability and accessibility of programming materials for children with visual impairments. *ACM Transactions on Computing Education (TOCE).*

Milne, L. R., & Ladner, R. E. (2018). Blocks4All: Overcoming accessibility barriers to blocks programming for children with visual impairments. *Conference on Human Factors in Computing Systems - Proceedings*, 1–10.

Morrison, C., Villar, N., Thieme, A., Ashktorab, Z., Taysom, E., Salandin, O., ... Zhang, H. (2018). Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities. *Human-Computer Interaction*, 1–49.

Prado, Y., Jacob, S., & Warschauer, M. (2021). Teaching computational thinking to exceptional learners: lessons from two inclusive classrooms. *Computer Science Education*, 1–25.

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

Read, J. C., MacFarlane, S., & Casey, C. (2002). Endurability, engagement and expectations: Measuring children’s fun. In *Interaction design and children* (Vol. 2, pp. 1–23).

Stefik, A., Ladner, R. E., Allee, W., & Mealin, S. (2019). Computer science principles for teachers of blind and visually impaired students. *ACM Inroads*, *10*(2), 50–57.

Taub, R., Armoni, M., & Ben-Ari, M. (2012). Cs unplugged and middle-school students' views, attitudes, and intentions regarding cs. *ACM Transactions on Computing Education* (TOCE), 12.

Tracy, S. J. (2010). Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative inquiry, 16*(10), 837–851.

Van Kesteren, I. E., Bekker, M. M., Vermeeren, A. P., & Lloyd, P. A. (2003). Assessing usability evaluation methods on their effectiveness to elicit verbal comments from children subjects. *Proceedings of the 2003 conference on interaction design and children*, 41–49.

Van Mieghem, A., Verschueren, K., Petry, K., & Struyf, E. (2020). An analysis of research on inclusive education: a systematic search and meta review. *International Journal of Inclusive Education, 24*(6), 675–689.

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching computer science to 5-7 year-olds: An initial study with scratch, cubelets and unplugged computing. In proceedings of the workshop in primary and secondary computing education, 55–60.

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *IDC Proceedings of the 2018 ACM Conference on Interaction Design and Children*, 289– 299.

Author details:

Anna van der Meulen, University of Leiden, The Netherlands.
Mijke Hartendorp, Saxion University of Applied Science, The Netherlands.
Wendy Voorn, Royal Dutch Visio, The Netherlands.
Felienne Hermans, University of Leiden, The Netherlands; Vrije Universiteit Amsterdam, The Netherlands.

Corresponding author: a.n.van.der.meulen@liacs.leidenuniv.nl