Physical Programming for Blind and Low Vision Children at Scale

Cecily Morrison[1], Nicolas Villar[1], Alex Hadwen-Bennett[2], Tim Regan[1], Daniel Cletheroe[1], Anja Thieme[1], Sue Sentance[2,3]

[1]Microsoft Research
[2]Kings College London, Centre for Research in Education in Science, Technology, Engineering & Mathematics
[3] Rasberry Pi Foundation

ABSTRACT

There is a dearth of appropriate tools for young learners with mixed visual abilities to engage with computational learning. Addressing this gap, Torino is a physical programming language for teaching computational learning to children ages 7-11 regardless of level of vision. To create code, children connect physical instruction pods and tune the parameter dials to create music, audio stories, or poetry. Currently, the uptake of novel educational technologies to support inclusive education of children with disabilities continues to be limited at scale. We consider how the Torino Learning Environment supports non-specialist teachers to teach computational learning to children with mixed visual abilities in a UK-wide evaluation with 75 children and 30 teachers over a period of three months. We demonstrate how children can successfully learn with a novel physical programming language. We articulate how key design constructs such as *persistent program overview* and *liveness* supported non-specialist teachers to co-produce learning for children of different ages, visual and cognitive abilities. We conclude with reflective guidance on evaluating inclusive educational technologies at scale.

CCS CONCEPTS

Human-centered computing → Accessibility → Empirical studies in accessibility

KEYWORDS

Inclusive learning; mixed visual abilities; blind; accessibility; tangible education technology; computational learning; evaluation at scale; computer science education (CSE).

Figure 1: Two learners with mixed visual abilities using Torino: a physical programming language for teaching computational learning to children ages 7-11 regardless of level of vision.

1 INTRODUCTION

Policy initiatives are being developed throughout the world to include programming, and computational learning more broadly (Cooper, Pérez, and Rainey 2010), into education and national curricula (e.g. (Peyton-Jones, Humphreys, and Mitchell 2013)). Addressing these initiatives, a range of specialist teaching tools have been designed and developed to encourage the development of computational learning in school children (Brennan and Resnick 2012; Cooper, Pérez, and Rainey 2010). A recent paper reviewed more than 47 different tools available for children between the ages of 2 and 18 (Duncan, Bell, and Tanimoto 2014). Tools for younger children are frequently block-based languages, such as the widely known Scratch (Maloney et al. 2010), but the momentum continues with the development and uptake of new physical technologies. Micro:bit, for example, was recently given to every 11-year-old in the UK (Rogers et al. 2017).

The formalization of computational learning into schools and curricula makes more apparent the lack of appropriate tools for many children with disabilities (Burgstahler and Ladner 2007). While there is growing research in the area of inclusive computing (Lechelt et al. 2018; Stefik, Hundhausen, and Smith 2011; Israel et al. 2015; Kane, Koushik, and Muehlbradt 2018), there remains a dearth of tools for teaching computational learning to children ages 7 – 11 with mixed visual abilities. Common languages used by their sighted peers, such as Scratch or Alice (Utting et al. 2010) are visual both in manipulating the code (e.g. drag and drop) and in the effect that the code has (e.g. animation). Existing physical programming languages also rely heavily on visual properties (Horn, Solovey, and Jacob 2008); distinguishing pieces, connecting them correctly, and experiencing the outcome of the program are all visual activities.

Addressing this gap, Torino is a physical programming language for teaching computational learning to children ages 7-11 regardless of level of vision (see Figure 1). To create code, children connect physical instruction pods and tune the parameter dials to create music, audio stories, or poetry. The design of the initial Torino prototype and early evaluations have been reported previously in (Morrison et al. 2018; Thieme et al. 2017). Multiple design iterations have resulted in a new, full-featured, manufacturable version of Torino. This is complemented by a scheme of work that guides non-specialist teachers and children through the concepts of the UK computing curriculum, forming the Torino Learning Environment.

Uptake of novel technologies at scale for children with mixed visual abilities in educational settings continues to be limited (Bouck 2016; Zhou et al. 2011). While researchers are addressing some of the challenges through co-design (Brule et al. 2016; Cullen and Metatla 2018), there remain challenges with the confidence and skills that qualified teachers of the visually impaired (QTVIs) have with technology. In the UK, the majority of blind and low vision children are in mainstream education leaving QTVIs expected to be fluent in all subjects, with the responsibility for adaptation (and often teaching) resting with additional teaching assistants (Metatla and Cullen 2018; Bach, Kessler, and Heron 2004). To ensure that specialist teaching tools for computational learning are utilized effectively in such settings requires their evaluation at scale.

We carried out a large-scale evaluation of the Torino Learning Environment with 75 children and 30 teachers situated across 24 localities in the UK. We provided Torino sets to teachers along with a scheme of work and teacher's guide for a full three-month academic term. We captured children's engagement and learning as well as considered the experiences of teachers through survey and diary instruments. Our findings show that children were highly engaged and that teachers reported age- and ability-appropriate learning across the cohort. Further, our analysis suggests that successful learning is not a simple end-point of the Torino Learning Environment, but something co-produced by non-specialist teachers.

This paper makes the following contributions:

- Demonstrates that children ages 7 – 11 with mixed visual abilities can successfully learn using a novel physical programming language at scale;
- Provides insight into design features, persistent program overview and liveness, that support or hinder non-specialist teachers to co-produce learning using the Torino Learning Environment with children of different ages, visual, and cognitive abilities;
- Offers reflective guidance for future research aiming to evaluate inclusive educational technologies at scale.

2 RELATED WORK

We set the scene by describing how children with mixed visual abilities are taught in the UK. We then discuss existing tools for computational learning and physical computing in educational settings. The final section discusses assessment of computational learning, motivating the evaluation strategy taken in this study.

2.1 Teaching Learners with Mixed Visual Abilities

In the UK, and most high-income countries, the majority of blind and low vision children are taught in mainstream schools (RNIB 2016). Learners are supported by peripatetic qualified teachers of the visually impaired (QTVI) who lead the adaptation process. In cases of severe visual impairment, a teaching assistant realizes adaptations on a day-to-day basis. There are also a very few specialist schools that work with blind and low vision students who have an additional or profound disability; local "special" schools for children with a range of disabilities unable to attend mainstream schools; and children schooled at home (approximately 10%). This variety suggests that a successful intervention needs to work across a wide range of settings and teachers.

The mainstreaming of learners with disabilities was motivated by a vision of social inclusion supported through inclusive teaching practices (UK Parliment 1981). In practice, inclusive education requires the use of multisensory teaching techniques so that all learners can participate regardless of their abilities. This stands in contrast to the model of assistance, either human or technological, that modifies materials to enable access, e.g. creating tactile graphics. Lack of resource and time on the part of teachers, coupled with new non-inclusive technologies in schools, have made the implementation of mainstream education for disabled children fraught with challenges, impacting academic and social participation (Gray 2009).

As a result, there has been a focus on addressing the (seemingly) more immediate need of making class materials accessible. This has been particularly challenging for STEM (Science, Technology, Engineering and Math) subjects (Moon et al. 2012). One approach has been to pair disabled and non-disabled students and divide the work by abilities -- a sighted learner manipulates the Scratch environment, while a blind child suggests ideas for an animation that s/he cannot experience. This however, leads to partial or even non-participation. Alternatively, teaching assistants (TA) are often asked to bridge the gap, making adaptations on the fly. This approach often leads to focused interactions between TA and student, creating an assistance bubble that isolates disabled children from the rest of the class (Metatla and Cullen 2018). Thus, there is a mismatch between the vision of social inclusion in education and the reality of teaching STEM subjects.

2.2 Inclusive Computational Learning Tools

Computational learning is being introduced into schools through a range of policy initiatives (e.g. (Peyton-Jones, Humphreys, and Mitchell 2013). Yet, there is much debate as to what it entails. We ground our theoretical approach in the work of (Cooper, Pérez, and Rainey 2010), who define computation thinking for school learning environments. They specifically highlight the iterative and interactive process between the student and the computer, making explicit the student's capacity for abstraction and for problem formulation. More practically, we draw upon three aspects of computation defined by (Brennan and Resnick 2012), which includes: Computational Concept, Computational Practices, and Computational Perspectives. Finally, we adhere to the learning goals of the UK computing curriculum for primary school children (Department for Education 2013).

A range of tools have been developed to support computational learning by learners with mixed visual abilities, summarized in (Hadwen-Bennett, Sentance, and Morrison 2018). The majority of these tools focus on making existing programming languages more accessible by making code structure apparent, e.g. (Baker, Milne, and Ladner 2015), or interoperable with assistive technology, e.g. (Ludi, Ellis, and Jordan 2014). Efforts have been made in *Quorum* (Stefik, Hundhausen, and Smith 2011) to ensure all learners can use the same platform through the creation of an evidence-based accessible language. The recent *Bonk* (Kane, Koushik, and Muehlbradt 2018) goes beyond an accessible language, actively engaging learners with mixed visual abilities to create audio games together. While most of these examples seek to support accessibility, only the most recent ones are designed to mediate an inclusive learning environment (Metatla, Thieme, et al. 2018).

To date, the development of computational learning tools has mainly been targeted at secondary school students (11+) competent in assistive technologies and with a fully developed working memory. There is currently no alternative to the Initial Learning Environments (ILE) used to teach primary school learners (6 –11) to code. Most ILEs are drag and drop block-based languages for creating digital media (e.g. animations). They offer abstractions of important programming concepts found in conventional languages, such as iterative or branching flow structures, without the complication of syntax. Recent work has documented the accessibility challenges of block-based ILEs and proposed a more inclusive alternative (Milne and Ladner 2018). Torino Learning Environment was designed to address the gap for learners ages 7 – 11. In the vein of inclusive education, it is intended to benefit all learners in a classroom regardless of visual abilities.

2.3 Physical Technology in Educational Settings

A good number of physical or tangible technologies have been developed for teaching computational learning: e.g. (Horn and Jacob 2007b; Zuckerman, Grotzer, and Leahy 2006; Sullivan, Elkin, and Bers 2015; Lechelt et al. 2016). Physical computing has been proposed to support human interaction. Evaluations of *Tern*, for example, highlight the advantages for child interaction when working on programs away from a computer (Horn and Jacob 2007a) as well as the opportunity to draw people into participation in classroom and museum (Horn et al. 2009). Video analysis of an early version of Torino (Thieme et al. 2017) as well as *Magic Cubes* physical toolkit (Lechelt et al. 2018) emphasize how the technology supports collaboration between people of different abilities, alluding to the consequent learning that can then take place. These references suggest that physical computing has the potential to support the kind of collaborative interaction desirable between children with mixed visual abilities.

Empirical validation of physical technologies for teaching computational learning in educational settings is rare (Zaman et al. 2012). Taking a qualitative action research approach, Virnes, Sutinen, and Kärnä-Lin (2008) consider how robotics can be used to teach hands-on programming with eight diverse learners with special educational needs over a period of nine months. A study with Magic Cubes uses interaction analysis to understand how collaboration, comprehension, and engagement are achieved when teaching computing concepts in a special school to 11 students age 16-19 (Lechelt et al. 2018). Wyeth (2008) takes a quantitative approach to observational data, capturing specified criteria that indicate program understanding, structure, debugging, planning, and multiple solutions. These studies focus on how the technology achieves learning goals through observation methods rather than measures of rate or breadth of learning across a cohort.

There are few studies at scale. Martinez, Gomez, and Benotti (2015) use multiple-choice tests to assess concept comprehension using a robotics platform in children ages 3-11 with 190 students. The data collection and analysis descriptions are limited so it is unclear how measurement was taken and whether there were multiple sites. Sentance et al. (2017) provide a qualitative study of the roll-out of the UK program to give a micro:bit to every 11 year old. Fifteen computing teachers along with 54 students in 8 focus groups (age 11-13) were interviewed. Despite specific prompts, teachers focused on engagement rather than characterizing learning and assessment. As such, there are no studies that attempt to evaluate a technology through assessing computational learning at scale.

2.4 Assessing Computational Learning

The lack of large-scale evaluation stems in part from the challenge of assessing computational learning. It is a research area in its infancy hindered by a lack of identified skills or competencies to measure (Giordano et al.

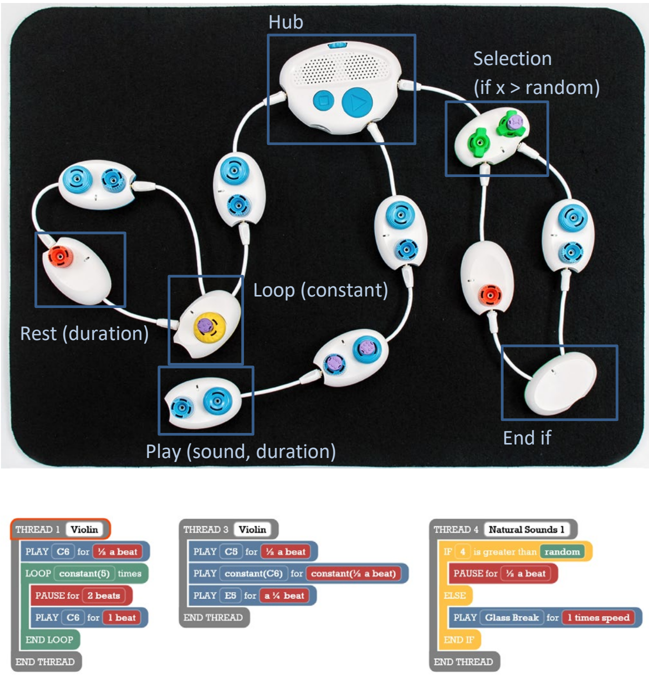

Figure 2: Torino system pieces connected in a multi-threaded program with text-based code below.

2015). A recent paper proposes a measured assessment of computational learning (Román-González, Marcos, Juan-Carlos Pérez-González and Jiménez-Fernández 2017); however, because it is highly visual and designed for learners age 11+, it is inappropriate to younger learners with mixed visual abilities. As an alternative, Brennan and Resnick (Brennan and Resnick 2012) propose a range of assessment activities: 1) analysis of constructs used in a program; 2) artefact interviews in which learners explain a program; and 3) programming exercises. Their combination is further proposed to enable a full overview of a learner's understanding (Grover, Cooper, and Pea 2014).

The assessment activities detailed in the literature all have challenges at scale. Analysis of constructs can be automated and easily done by a non-specialist teacher, but it has been shown that usage does not equate to code understanding. Further, achieving this through log data analysis requires an individual log-in for each child, which is impractical when working with a physical system, (non-specialist) teachers, or in groups. While artefact interviews better address understanding, they are time consuming and require teacher expertise to formulate prompt questions. Programming exercises can be set by researchers but are best used as a formative assessment as the line between assessment and learning is unclear. As such, there is currently no scalable measurement instrument of computational learning in the education literature that is: accessible to children; accounts for shared technology use; and usable by non-specialist teachers (Kallia 2017).

We can, however, measure motivational and engagement constructs. The Expectancy-Value Model (Eccles and Wigfield 2002) of motivational constructs theorizes that both expectancies and values influence career choices and are a better predictor of career choice than attainment in primary school. Engagement is also a key part of the learning process and can be measured through a validated research instrument in computing students ages 5 to 18 (DeLyser, Mascio, and Finkel 2016).

3 TORINO LEARNING ENVIRONMENT

The Torino Learning Environment is a combination of the Torino system (hardware and software), the scheme of work (lessons), and the approach taken in introducing these to teachers (set provision).

3.1 Torino System

Torino is a physical programming language for teaching computational learning to children age 7-11 regardless of level of vision. To create code, children connect physical instruction pods and tune their parameter dials to create music, auditory stories or poetry as shown in Figure 2.

Program Flow

Each pod is a statement in the program. Learners can build up a range of program flows with different pods. In this new version of Torino, in addition to play, rest, and loop pods, there is also *selection* (if then), and *merge* (end if). Each pod has a number of connectors and cables that allow them to be plugged together to define the structure of the program. Pods plug into one of four jacks on the main unit (Hub), representing the logical starting point of the program with each connection point the start of a thread. For example, Figure 2 shows a program with three concurrent threads. The first thread will play a single sound, and then will enter a loop that will repeat the action of playing a rest (silence) followed by a sound. A second thread will play three sounds in sequence. A third thread will play either a sound or a rest, depending on the outcome of a conditional statement.

Figure 3: Two blind children looking at the variable they have created. The child on the left holds the second "read" variable to be put in the second loop.

Torino was deliberately designed to have a low floor and high ceiling. Seven year olds (or those with additional learning needs) can start with very simple programs: a sequence of three play pods to learn that a program is a sequence of commands; whereas 11 year olds can combine constructs, such as nested loops or looping selection (if then) statements, to push their understanding of program execution. Each Torino set has 15 pods (8 plays, 3 rests, 2 loops, selection, and merge) keeping the focus on the constructs and their execution, rather than long programs. The type and number of pods was determined to ensure that our matching scheme of work could cover all concepts in the UK national curriculum for this age group.

Each pod was designed to be tactually and visually distinct. The base shell of the pod is the same while the top has different slopes and textures along with differentiated placement and number of dials. Dials each have a distinct texture and are colored to support those using visual information. This includes sighted children and teachers, as well as many blind and low vision children. We were careful to avoid disparity between tactual and visual information to ensure unimpeded interaction between those of different abilities.

Data Flow

Pods have knobs that represent configurable parameters, which can be rotated to specify the value of the parameter. Play pods have two knobs, one to specify the sound and one duration; rest pods have a single knob for the duration of silence; the knob on a loop pod specifies the number of loop iterations; and the pair of knobs on a conditional pod represent the values of the conditional statement x > y. Dials have eight possibilities before looping (e.g. 1,2,3…8,1) with the exception of duration which has four. New to this version of Torino are plugs that can be inserted into the knobs to *programmatically* change their values, enabling taught concepts to include constants and variables. The set of plugs include: constant values 1-8, random, infinity, increment and decrement counters as well as variables. Variables can be assigned by placing a plug into a variable sleeve and then into a dial. The variable is read by placing an empty variable sleeve into another dial as demonstrated by the children in Figure 3.

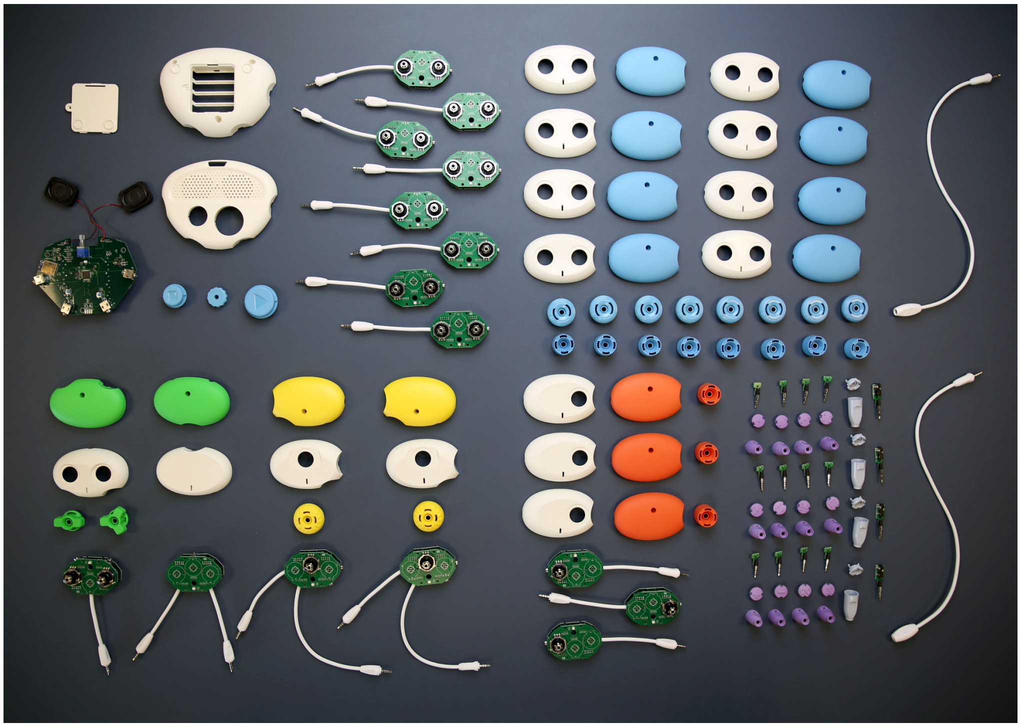

Figure 4: A picture of the individual components in the Torino set, including 105 plastic parts, 48 circuit boards, and three custom cable designs.

Twisting a dial gives an immediate audio response if a pod is plugged in. Children can rotate through their options until they find what they are looking for. This *liveness* was a specific design feature used to mimic the liveness of ILEs (Tanimoto 1990; Burg, Kuhn, and Parnin 2013). It was derived from our iterative design process that highlighted the ways children engaged with the world through their hands (Morrison et al. 2018). Pods hold their state if unplugged and re-plugged into another thread on the Hub.

The sound set of each thread can be changed by selecting it in the software. Torino currently includes midi instruments as well as samples of natural sounds, e.g. siren. Poems and stories can also be recorded and broken into sections to work with. With some planning, teachers can add student sounds to the system.

Reviewing Code

Students are expected to read their code physically. They are particularly encouraged to follow their program as it executes, precisely touching or pointing to each pod as the program progresses. Research has noted that incorrect mental models can form when there is an inadequate understanding of the 'hidden' processes that are not directly observable from the program (Sorva 2013). As a result, Torino followed the design construct to provide a *persistent program overview* of the program at all times. Combined with the physicality of the program, this design approach encourages computational learning through planning and prediction (algorithmic design (Waite et al. 2018)), and by following program execution (tracing and debugging (Lister et al. 2004)). Physical program following has the added benefit of supporting shared attention between learners and can assist in debugging as the learner's hand is already in position to fix a bug when spotted.

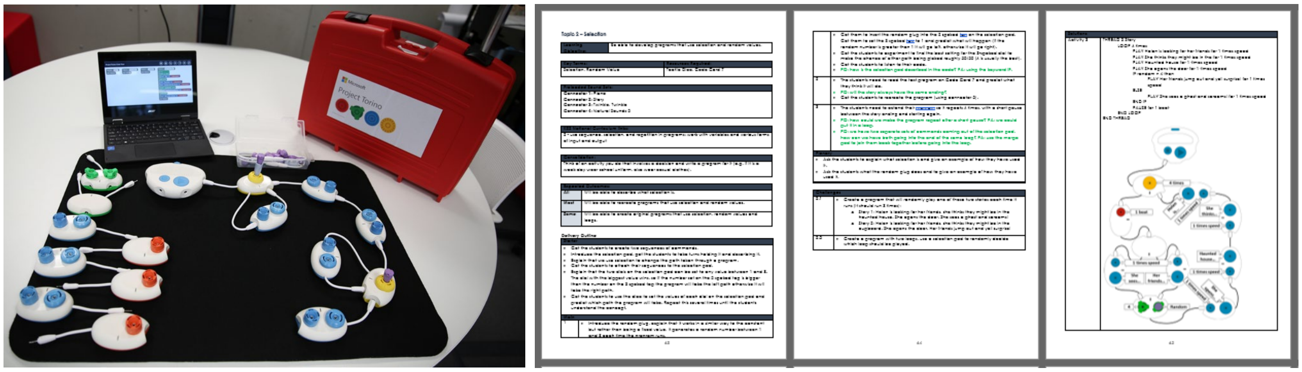

Figure 5: (left) Set as provided to teachers includes Torino kit and pre-paired tablet; (right) example from the teacher's guide which includes differentiated outcomes, activities, prompt questions in green, and a pictorial description of the answer to the challenge.

As students get ready to transition from Torino to a text-based language, they can listen to their code by pressing simultaneously the play and stop button on the Hub. Those with vision can view their code in software in an appropriate visual medium. We found teachers and adults often used the visual code (see Figure 2).

Hardware Description

Each pod contains a custom-designed circuit board, a microcontroller and connectors which power connected pods and enable them to communicate. Messages, including type of pod and current state, are propagated through the network until they reach the Hub. From these messages a network graph is constructed, where a node is a pod and the edges are the connections between them. Plugs also contain a custom-circuit board design attached directly to a connector that can connect through the dials to the pod circuit board. The processing is done on a linked device, e.g. tablet.

Design for Manufacturer

A substantial part of the design iteration focused on capturing the design traits of the first prototype (see Morrison et al. 2018) in a form that was manufacturable at scale. Two key changes included a change to a single configurable circuit-board for all pods and a change to a single interaction mode of dials. As a result, all pods had the same basic shape (and base shell) and are tactually distinguishable by number and placement of dials as well as differently shaped upper shells. Nonetheless, the final kit includes: 45 injection-moulded part designs, 5 circuit board designs assembled in 25 different configurations; and 3 custom cable designs. The final kit has 105 individual plastic parts and 48 circuit boards. See Figure 4.

3.2 Scheme of Work

A scheme of work was developed to support non-specialist teachers using Torino. It was our assumption that Torino would be used by QTVIs, teaching assistants, and parents. We expected none of these people to have programmed previously. The scheme was developed by a researcher in computer science education with substantial classroom experience. Topics cover the learning goals of the UK Curricula for children ages 7 – 11 (Department for Education 2013). Learners are expected to "design, write and debug programs that accomplish specific goals" and solve problems through decomposition. They are expected to do this by using: *sequence*, *selection*, *repetition*, *variables*, and *inputs and outputs*; and to *use logic to detect and correct errors in algorithms and programs*.

The scheme of work that we developed contains 12 topics broken into two modules as well as five unplugged activities. The first module covers: sequences, parameters, threads, debugging, loops, loops with sequences, and problem decomposition. The second module contains: constants, selection, variables, counters, and nested loops. Unplugged topics include: understanding computing careers, binary numbers, and networks. Each topic is presented as an interactive learning resource that can be worked through independently by older learners or presented to the learner by the teacher. It begins with new vocabulary (e.g. debug) followed by a series of exercises. A use-modify-create strategy was utilized to introduce concepts (Lee et al. 2011). Code cards with pseudocode are given to learners to create with pods and then modify. Each topic finishes with one or more challenges. Challenges can be complete programs that the learner must listen to and recreate or open-ended instructions to inspire creative programming.

There is an accompanying comprehensive, 100+ page Teachers' Guide shown in Figure 5(right). It includes differentiated outcomes, activities, prompt questions in green, and a pictorial description of the answer to the challenge. We specifically encourage teachers to ask questions to stimulate thinking, e.g. *"How many commands are in the program that you just listened to?"* We emphasize planning (algorithm design), predicting program output, and iterative debugging. We also stressed use of appropriate language (Grover, Cooper, and Pea 2014).

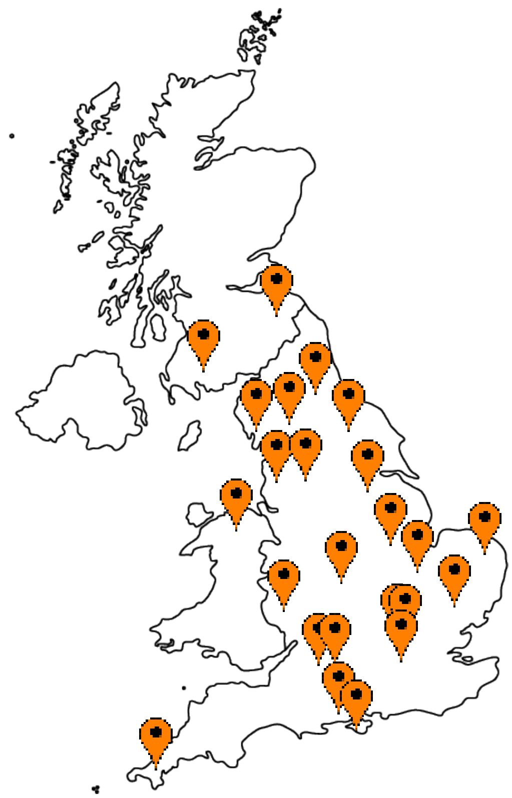

Figure 6: Map of participating educational settings.

3.3 Set Provision

Each set included a Torino kit and tablet as shown in Figure 5 (left). Sets were prepared to ensure the out-of-the-box experience was not frustrating for teachers: the tablet and Torino Hub were pre-paired through Bluetooth; the software was pinned to the task bar; and the learning resources pre-loaded and pinned to the task bar of the tablet. Internet in the educational setting was not required. Teachers also received a Frequently Asked Questions (FAQ) document that highlighted available accessibility features. A teachers' email list was available, and the researchers answered questions sent to the project alias. Weekly top tips were also provided to encourage the teacher's thinking.

Training was provided to all participant teachers. Those with more than one learner were trained individually, mainly in person, although some remotely. Those with a single learner were trained through group workshops in London and Manchester, attending both a kick-off and closing workshop. Each teacher was assigned a mentor who was a blind or low vision person working in the computing industry. Teachers were asked to carry out a minimum of 8 hours of instruction for each student over the period of a school term. No specification was given about the format of the sessions as we assumed educators would need to fit within the constraints of a range of educational settings. We encouraged teachers to have students with mixed visual abilities work as pair programmers (Sentance and Csizmadia 2016).

4 STUDY DESIGN

The Torino Learning Environment aims to enable learning for children with a range of abilities supported by non-specialist teachers in varied educational settings. From the literature we identified: (1) limited uptake of novel technologies at scale for children with mixed visual abilities (Bouck 2016; Zhou et al. 2011); and (2) challenges of up-skilling non-specialist teachers at scale (Sentance and Waite 2018). To judge whether we successfully addressed these issues, we ran a large scale evaluation of the Torino Learning Environment with 30 teachers supporting 75 children and across 24 localities in the UK for 3 months illustrated in Figure 6.

We ask the following research questions:

- Were teachers invested in and able to mediate learning with the Torino Learning Environment?
- Did students learn using the Torino Learning Environment?
- What design features of the Torino Learning Environment enabled non-specialist teachers to facilitate learning?

4.1 Data Collection

We designed a mixed-method study that captured different angles of student learning. In our study design, we were sensitive to the following key characteristics of our setting:

- students are diverse in their abilities, so we cannot expect a single learning end-point for all students;
- teachers are non-specialists and busy;
- data collection by researchers (e.g. video or skilled assessment) is restricted due to logistics of scale and school entry by non-teachers.

The data set includes a combination of semi-structured questionnaires, open-ended diary entries, and technology use logs. These different data sources enable data triangulation of learning while meeting the above criteria. This approach addresses the lack of an appropriate summative measure of computational learning that could be used without direct researcher supervision by focusing on teacher reported learning. It also recognizes that given the diversity of students and a lack of a learning end-point, teachers are best placed to articulate their students' learning. We report data from three research instruments as relevant to the above research questions: Learner pre/post questionnaires, teacher questionnaire, and teacher diaries. We do not include technology use logs as we found looking at decontextualized use, given multiple users of the system, told us little about student learning.

Learner Pre/Post Questionnaire

A learner questionnaire was given pre and post Torino usage to understand learner engagement and motivation. It consisted of 8 and 12 questions respectively with a mix of Likert scale and free response questions drawn from a variety of sources. We report here three pre/post questions measuring change in motivational construct (self-efficacy (Kasanen, Raty, and Eklund 2009), reported interest, and perceived future usefulness). Three post only questions are intended to measure engagement with the Torino Learning Environment (excitement, perceived importance, and frequency of thought) taken from a validated questionnaire to measure engagement with computing for school students (DeLyser, Mascio, and Finkel 2016).

The self-efficacy question asks learners to give themselves one to five stars. All others are discrete Likert scales taken In line with typical Likert scale formats, "1" equates to strongly disagree and "5" to strongly agree. However, the wording is adjusted for each question. For the question, "How useful do you think coding / programming will be to you in the future?" students can tick the box for: Not at all useful, slightly useful, somewhat useful, quite useful, and extremely useful. Materials were created to be as accessible as possible either through enlarged text or through a screen reader. However, it is not uncommon for teaching assistants to read out and fill-in paper documents for learners who are braillists.

Teacher Post Questionnaire

A teacher questionnaire was given post Torino Learning Environment usage, intended to capture a quantitative snapshot of its benefit. It included five Likert scale questions and a section for free response comments. Questions included desired future use as well as probed the challenges around understanding coding concepts or using the scheme of work. Teachers were instructed to answer each statement on a scale from 1 to 5 (1 = disagree strongly and 5 = agree strongly).

Teacher Diary

The teacher diary provided a structured way for teachers to reflect on the learning that was, or was not, taking place with the Torino Learning Environment. Teachers were asked to focus on three areas: 1) aims and whether they were achieved; 2) engagement and disengagement; and 3) logistics of teaching sessions, including adaptations made. Diaries were requested after the 4th and 8th lesson. These diaries were deliberately designed as more open-ended inquiries, specifically intended to help us understand what kind of learning took place without specifying targets.

4.2 Participants

Participants were recruited to use the Torino Learning Environment through an online blog post and a demo booth at a UK-wide conference for QTVIs. Teachers (including parents) could either apply to support a number of children (e.g. across a Local Authority) or a single child. Torino sets were allocated to all those who applied and fit the criteria for age and location.

Qualitative Data: The qualitative data in the paper is drawn from **30 teachers** supporting 75 students across 24 localities who provided at least one teacher diary. Sixteen of these teachers were QTVIs or teaching assistants, six were parents, five were IT or resource coordinators, and three were specialist IT teachers. Sites were spread across the UK from Cornwall in the southwest to Edinburgh in the northeast, spanning all social demographic levels. We included all data received to ensure we captured any negative views that may have stopped participation or data submission.

Quantitative data: Quantitative data was included from 59 **learners**. Given a stronger need for completeness to interpret the results, we only included learners if both pre and post questionnaires were returned and the learner had engaged in the minimum 8 hours of use. The demographics of the quantitative data is shown in Table 1. A wide range of visual abilities were covered with about a third having an additional disability. There was a skew of boys to girls. Survey data was returned from **22 teachers**.

Attrition: A total of 112 **learners** were originally allocated 50 sets. Attrition mainly came from two large sites requesting large numbers of sets and then dropping out before sets were received (20 learners). Another large center (8 learners) lost their IT coordinator and therefore did not have capacity to submit research data even though the sets were used. The remaining attrition was a mix of changes to school setting (2 learners), health deterioration (3 learners), or inability to attend training (4 learners).

4.3 Data Analysis

The data of the learner and teacher questionnaires was not normally distributed. Thus, a Wilcoxon Signed rank-test is used to compare pre and post data to understand changes to motivational construct. Descriptive statistics are used to present post-only engagement data. Mann-Witney U tests are used to consider significant differences

Table 1: Participants for quantitative data by gender, vision level, additional disability, and educational setting

Gender		Vision Level		Additional Disability		Educational Setting	
Female	28.8% (17)	Blind	35.6% (21)	None	67.8% (40)	Mainstream	57.6% (34)
Male	62.7% (37)	Blind (Residual Vision)	28.8% (17)	Learning	11.9% (7)	Other	33.9% (20)
Unknown	8.5% (5)	Partial Sight	27.1% (16)	Physical	6.8% (4)	Unknown	8.5% (5)
		Unknown	8.5% (5)	Other	5.1% (3)		
				Unknown	8.5% (5)		

across sub-group types: gender, vision level, additional disability, and type of educational setting. Descriptive statistics are also used to report the teacher questionnaire data. In all cases, median values are presented because the data is not normally distributed.

Thematic analysis was used with the teacher diaries (Braun and Clarke 2006). As responses were highly varied, some answering the questions and others reflective of the teacher's interest, two researchers individually identified episodes of learning and reflections on teachers' experiences. These were then cross-checked and any differences discussed and resolved. Learning episodes were then thematically group using Brennan and Resnick's framework of computational learning (Brennan and Resnick 2012): 1) *Computational Concepts* (e.g sequences, iteration, loops, selection, variables); *Computational Practices* (incremental & iterative, testing and debugging, problem decomposition); 3) *Computational Perspectives* (expressing themselves with computation, connecting with others through computation, questioning the world around). Further themes were drawn out around two key design features of Torino: *persistent program overview* and *liveness* (Morrison et al. 2018).

5 RESULTS

The majority of teachers reported using the Torino Learning Environment weekly for 45-75 minutes, scheduled when convenient rather than in conjunction with existing computing lessons. Below are two example lessons from the same teacher. The first is for an able 9 year old girl who is blind with residual vision and the second is for a 7-year old boy with no vision who may have additional learning needs.

> *The last session I had with the year 4 pupil was Threading, she was able to create 2 sequences of commands. When asked what would happen when she ran the program she replied "It will play two at the same time." However, when she ran the program thread 1 did not run so she went from play pod to play pod checking their connections. She then ran the program again and this time both threads played. I introduced the pupil to the pause pod. I then played her the example program she was able to state that there were two threads, she was also able to say how many commands were used. She then recreated the program for thread 1. She then tried to recreate the program on thread 2, however, when she ran the program she noticed that thread 2 was not correct and realized that she had not counted the pause as a command. She then began to reconnect the pods and ran the program, tracing thread 2 as she listened and made adjustments to fix any errors. She was then able to run the program successfully. (T5)*

> *His last session – ask him to get out the hub and pods and set up a train of pods to make the ambulance noise he loves. Then move onto setting up 2 threads and trying to match the sound example of the story with sound effects using the pause pods. Once achieved, add a bug to one of the threads and ask him to find it and fix it. He then wanted to put bugs in for me to find with his help. Finish by setting up so he can listen to the ambulances again. Student then puts all the equipment away with minimal help. (T5)*

These two descriptions illustrate the diversity of children's learning needs that teachers needed to address with the Torino Learning Environment. With this context, we first consider the experience of teacherss as mediators of the learning experience. We then look at learning as captured through quantitative measures and teacher reports.

Table 2: Median results of the Teacher Likert-scale Questionnaire with 1 = disagree strongly and 5 = agree strongly. To illustrate the positive response of teachers, we have included the ideal response in brackets.

Question	Median Response
I think Torino is a good tool for teaching coding to visually impaired children.	5 (5)
I found some of the computing concepts hard to understand.	2.5 (1)
Teaching with Torino helped me to improve my own computing subject knowledge.	4 (5)
The teachers' guide was hard to follow.	2 (1)
I would like to use Torino to teach coding in the future.	5 (5)

Finally, we bring both of these perspectives together to reflect on Torino as a tool designed to support learning.

5.1 Teacher Mediation

We first consider the experiences of educators, asking the question: "Were teachers invested in and able to mediate learning with the Torino Learning Environment?" This is an important foundation to learning for children.

Teacher Aims

Teachers generally had positive views as illustrated by their rating of the Torino Learning Environment as a good tool for teaching children who are blind or low vision and their desire to use it again as shown in Table 2. This aligns with the aims for using Torino that teachers expressed in their diaries. Most teachers wanted to help their learners understand code, with some framing that particularly as a matter of access. One teacher portrayed the situation through the following quotation:

> *For my 2 students to be able to access programming ideas and principles, including problem solving and thinking through sequences in a logical way. Neither student can use Scratch or similar KS2 programs due to their VI needs so do other work during these IT whole class sessions or work with a partner who then takes the active role in making any decisions. (T9)*

Indeed references to "access" by teachers articulated this as both a need for opportunity to gain the problem-solving and logical thinking emphasized in computer programming as well as address the social impact on the students of being excluded from a subject in school. From the survey results presented in Table 2, we can infer that teachers were satisfied that their aims of accessing and learning were met by the Torino Learning Environment.

Non-specialist teachers

The challenge of teaching with Torino as a non-specialist also seemed manageable in most cases. The survey results suggest that the Teachers' guide was usable but with some room for improvement. The question of whether computing concepts were difficult to understand had the widest spread of answers as some teachers found this a real challenge, while others were specialist teachers with existing subject knowledge. Teachers with a math or science background generally described the material as easy to engage with. A number of teachers spoke about their own personal teaching journey with Torino as exemplified by the quotations below from a teacher's first and second diary entry.

> *Initially I was slightly apprehensive as I don't see myself as understanding code. (Diary 1, T8)*

> *Now I am more confident with the equipment and the concepts, I am more aspirational for the students. (Diary 2, T8)*

Teachers noted strategies of working together with a colleague (although that too posed challenges for peripatetic teachers) and allocating extra time to learning the "trickier" aspects. One teacher mentioned that a focus on the language helped provide a concrete approach to a subject that was unfamiliar. These approaches seemed to give teachers confidence, yet they require the expectation that time can be allocated to teacher learning. With purpose and confidence, we can expect the teachers to be positive mediators of the learning experience.

5.2 Learning with Torino

We next report data to answer the question: *Did children learn using the Torino Learning Environment?* We consider: 1) engagement; 2) motivation for further pursuit; and 3) teacher-reported learning.

Engagement

Table 3: Results of the post-only questions in the Learner Questionnaire with 1 = negative response and 5 = positive response. To illustrate the positive response of learners, we have included the ideal response in brackets.

Engagement	Median Response
How excited are you before you start your Torino sessions?	5 (5)
How important is it to you to do well in your Torino sessions?	5 (5)
When not using Torino, how often do you talk about ideas from your Torino sessions?	Never:17 Monthly: 7 Biweekly: 8 Weekly: 23 Daily: 4

Students reported strong engagement as measured by excitement levels before a Torino session and the importance of doing well in a Torino session as shown in Table 3. There was a range of responses to how often learners thought about Torino outside their lessons with a peak around weekly. There was little difference between sub-groups. Participants in mainstream schools tended to rate their excitement before Torino sessions and the importance of doing well more highly (Median = 5) compared to participants in other provision (Median = 4) ($U=243$, $p = 0.048$ and $U=234$ $p=.035$ respectively). While engagement is a prerequisite for learning and one of the most common approaches to assessing new technologies for computational learning, motivation to continue to learn is thought to be more indicative of future learning.

Motivation for Further Pursuit

Table 4: Results of the pre /post questions in the Learner questionnaire with 1 = negative response and 5 = positive response.

Motivation for Further Pursuit	Median Response		
	Pre	Post	P-Value
How many stars would you like to give yourself now for your coding / programming ability?	2	4	P<.0001
How Interested are you in coding / programming?	4	4	P=.160
How useful do you think coding / programming will be to you in the future?	4	4	P=.240

Self-efficacy in coding ability grew substantially after exposure to the Torino Learning Environment with a change in the Median score from 2 to 4 with large effect size, $r = -0.730$. This comes in the context of learners rating coding as interesting and personally useful: Median = 4 in both cases. Learners not in mainstream education showed an increase in interest from 3 to 4 (Median) with a large effect size of $r = -0.512$ and significance level of $p = .022$. These results are shown in Table 4.

Teacher Identified Learning

Computational Concepts

All sites indicated that students had learned a range of computational concepts. This was expressed in a number of ways. First, teachers described programs that students had created themselves:

> *In the last Torino session, the two higher-ability learners were creating their own tunes using the piano sounds and making use of loops, nested loops, pauses and variables. They enjoyed having the freedom to try out what they had learned previously. (T4)*

Second, students demonstrated their knowledge verbally by explaining the utility of a particular piece and giving an example:

> *He worked through the tasks and at the end could explain to me what a variable was and an example of how he had used one. (T20)*

Third, students used the correct vocabulary, which many teachers rightly understood as being key to ensure that learners could interact with other students not using Torino:

> *When completing activities, the children now often use correct key terms – 'sequence', 'thread', 'parameter' etc. (T10)*

While the endpoint of learning varied with the age and ability of the student, all deployment sites reported learning of concepts and vocabulary across the cohort.

Computational Practices

A majority of teachers mentioned problem-solving as a key area of learning. Some focused on the *"automatic"* response learners developed to *"work out"* or *"fix"* their programs:

> *The most noticeable impact on progress has been the development of problem solving skills. During the first handful of Torino sessions, the children struggled to identify where to start when repeating an example task. Now, they are quick in identifying roles for each other, tracing and building the sequence of code. (T1)*

Other teachers suggested that problem-solving was profound in changing students' beliefs about how they approached learning. One pupil "now believes they can 'ace' a challenge – even if it requires a bit of debugging first," suggesting that confidence has been gained. Another learner has become more patient now that he can name the process he must follow if something does not work – an experience noted for various students who showed autistic spectrum condition learning traits.

Problem-solving, although most commonly related to debugging by teachers, also included a range of other computational learnings. The data provides examples of both decomposition and efficient programming. The following example describes the latter:

> *She corrected her program when she realised she didn't have enough pods to complete what she originally wanted to do…thought about possible solutions and decided to only have one sound towards the start of the story when the main character appears…Through trial and error, she worked out where to place the pause pods in both of her threads and was successful. (T3)*

While teachers may have been less aware how these examples connected to computational learning, that several such examples were posed, illustrates their importance to teachers.

Computational Perspectives

A few teachers explicitly mentioned inclusive learning. In some cases this was a main focus of the teaching and success was reported in strengthening *"collaborative problem-solving skills, listening skills, and solid friendships"*.

Other teachers reported students explaining programs and code cards to friends and family as a way of sharing (and reinforcing) their learning. Teachers noted the important role of vocabulary in making this a shared experience with other students.

Stories of Learning

A large number of student specific learning examples were raised that spoke to the general development of the child rather than computational learning *per se*. To give two examples:

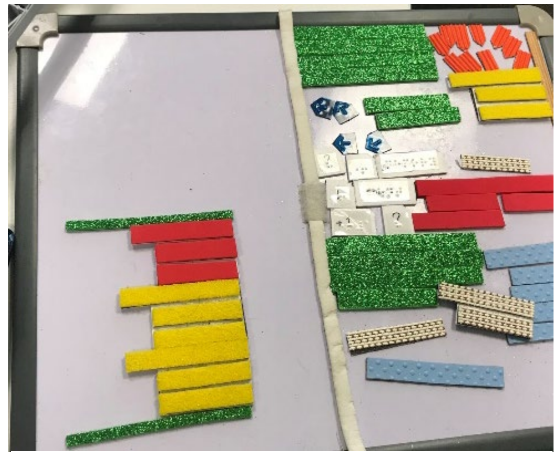

Figure 7: Tactile representation of code created by a teacher to support transitioning to text-based code.

> *Bobby is blind with an additional autism spectrum condition. He attends a special school. His qualified teacher of the visually impaired (QTVI) noted that he had uncharacteristically good concentration when working with Torino and could work for an hour at a time. The teacher noted that Bobby started "using the debugging process to solve problems that he encountered (non-torino based) between sessions, showing that this was a good way to help him develop generalised skills".*

> *Jeremy is a blind boy, age 10, with a very strong interest in technology and exceptional maths skills. He attends an independent school with small class sizes, but struggles to engage with the other children due to lack of a good shared activity. Jeremy found that Torino could be a shared activity with meaning with his sighted classmate that could be completed without the interference of a teaching assistant.*

These extra-curricular benefits can be extremely powerful for these children and are an important part of the efficacy of the experience.

5.3 Reflections on Design for Learning

We consider how two key design constructs supported non-specialist teachers in enabling the above described learning.

Persistent Program Overview

Program Following (Tracing)

The most important design construct was to have a persistent overview of the program, an extreme contrast to programming with assistive technology which requires substantial working memory to contextualize the line of code seen or heard. The importance of a persistent program overview was found to be particularly useful in activities of problem-solving and debugging. It was a consistent theme across teacher diaries, in which several teachers made explicit references by speaking about students' abilities to follow or trace program execution:

> *I should mention that we made good use of the tip relating to 'tracing' the programme. For some students this is really important and aids their understanding. The students felt it and described it. (T18)*

Others mentioned it as part of the learning process, such as hand-over-hand joint learning, something commonly done with children who are braillists.

Extensibility

Requiring a persistent overview limits the complexity of the programs. However, we were able to build in extensibility by allowing teachers to change the sound sets. This proved critical for teachers to adapt to the heterogeneity of their students' abilities. For example, one school focused on "creating and debugging lyric based songs or stories" as it was motivating. Another parent created many activities with loops for her child. A teaching assistant created simpler programs for her student, making them herself and recording them on the mobile phone. She noted that this was a very successful teaching approach for her student, but required accommodating her mobile phone in lessons which was distracting.

Seeing the bigger picture

Some teachers found it difficult to help their learners relate the coding they did with Torino to their other experiences with technology:

> *The greatest challenge to the aim of developing the children's understanding of computer coding is that some of the group still struggle to link the activities they are completing to how computer code works in the 'real world'. (T12)*

Torino's design supports learners in linking pods to lines of code by reading out or viewing the code. This feature was missed by many teachers as it was not immediately apparent (despite two mentions in the top tips). Some teachers created their own materials to support this connection as shown in Figure 7. Notwithstanding the creativity of teachers, the difficulty of relating code to computers is a challenge for any non-specialist teacher.

It is indicative of the more general challenge of understanding the specific, desired learning outcomes. We saw that teachers struggled to determine appropriate expectations for individual students. One teacher, for example, was disappointed when a blind child age 7 (likely with additional learning needs) could only create programs of 3 pods long. This was put across as a failure of system and student, rather than a potential very valuable learning for a student with a very weak sense of sequence (very common in young blind children). Supporting differentiation of learning more explicitly could give teachers more confidence.

Liveness

The liveness, or immediacy of Torino responding as soon as manipulated, was consistently mentioned as key to engagement. This is eloquently captured by one of the specialist IT teachers at a school for the blind with a deep knowledge of accessible programming:

> *Currently entry level environments such as Scratch are either inaccessible to my students or provide very dry feedback i.e. text based output that is then read using a screen reader. The same output could easily be achieved by writing in a text editor. The perceived relevance of programming can be lost because of this. One of the advantages of a product such as Torino is that it provides immediate feedback to students from the very first plugging in of a 'play pod'. The physical nature of the device removes some of the abstraction of creating and running a programme using an IDE. (T17)*

The immediacy of feedback also encouraged free experimentation:

> *Disengagement looks like randomly connecting pods midsession and then questioning what that means or does; an excellent way of learning but it sometimes means that we have to step ahead a few steps and then pull them back again. (T8)*

Iterative experimentation is often encouraged in computational learning (Brennan and Resnick 2012), but it can make it difficult for non-specialist teachers to support learning as they may not have "the answers" to student questions. In contrast, the majority of teachers did feel confident adjusting tasks or the pace of learning to suit individual needs.

Inclusive Learning

Some teachers felt that the liveness was so important that they were disappointed when students had to work together:

> *I haven't been aware of any disengagement. The only issues have been around turn taking because of not having access to an individual system.*

Teaching students separately does not take advantage of the collaborative learning possible with Torino (Thieme et al. 2017). In contrast, a minority of teachers centered their teaching around inclusive learning. One teacher, for example, focused explicitly on Torino as a mechanism for shared interaction between four children from different schools:

> *Joint story creation exercise – each pupil was given a real life cuddly toy which they had to describe to their peers. These then became the characters in a [dynamic story] that the pupils created [and coded] thus encouraging the use of descriptive language and developing their listening skills. (T9)*

The same teacher commented at the close of the study: *"Torino has provided an equal playing field for the pupils to be able to learn the skills, concepts and language required to progress through the scheme together."*

6 DISCUSSION

This paper reports a successful large-scale evaluation of a physical programming language for teaching computational learning to children ages 7 – 11 with mixed visual abilities. Over a period of three months, 75 children and 30 teachers worked through the scheme of work within the Torino Learning Environment. Non-specialist teachers were able to adaptively support the learning of children as illustrated by the range of data collected: high levels of student-reported engagement and motivation to continue study as well as consistent teacher reported learning.

This was achieved using two key design features of the language: its *persistent program overview* and *liveness*. The physical persistent representation of the code enabled teachers to support students in demonstrating their understanding of program execution by physically following their program, while the extensibility of the software allowed teachers to quickly adapt content to support engagement. The liveness generally encouraged engagement, but also raised challenges for non-specialist teachers to manage unexpected questions and distractions (e.g. phone).

Completing a large-scale study of an educational technology that aimed to: 1) assess learning with a novel technology; 2) work with non-specialist teachers; and 3) create an inclusive learning experience, is not without challenges and trade-offs. We reflect on these below.

6.1 Assessing Learning

As a result, in this study we relied heavily on teacher-reported learning, requiring non-specialists to assess computational learning. Our data demonstrated that teachers had the intuition to assess learning using the materials provided (e.g. quizzing on vocabulary). However, the heterogeneity of their reports suggests that teachers were uncertain about what constituted computational learning. This impacted reported learning in a variety of ways. Some teachers only reported programming constructs learned, missing the broader aspects of computational practices and perspectives. Others missed student appropriate learning that we surmised from reported activities as it was more basic than teachers may have expected. Our findings suggest that it is important to scaffold teacher's understanding of potential learning outcomes throughout explicit frameworks.

In retrospect, we needed to do more to scaffold teachers understanding of computational learning for assessment purposes. Specifically, we could build on the approach of program explanation (Grover, Cooper, and Pea 2014) and the physicality of the language. Teachers could assess learning by asking students to physically follow their program during execution and then repeat with an explanation of the program. This could be supported with a number of teacher-specific tools: an auto-generated explanation of the code to judge correctness and an easy mechanism to share student code and explanations with expert teachers to support upskilling. Finally, a framework of targeted skills is needed to ensure teachers include a broad range of learnings and differentiate appropriate levels of learning for individual children.

An important, and unexpected, result of open-ended teacher diaries was frequent reporting of learning that focused on the whole child rather than just computational learning. Teachers described the development of joint working skills or sustained attention. Had we used a summative assessment or had a very structured assessment protocol, we might have missed the powerful extra-curricular learning experiences that students had. These may be particularly important to children with disabilities who might have had developmental impact or limited participation in non-academic spheres. We suggest that researchers invite and value the reporting of whole-child learning.

6.2 Working with Non-specialist Teachers

Teachers, despite being non-specialists, showed substantial ingenuity in adapting Torino to support the motivation and learning of their students. Many created their own sound sets based on students' interests and abilities as well as own program challenges. This willingness to make and adapt by QTVIs has been highlighted as a "maker culture" [4]. More could be done to support this making. For example, the Torino software could enable the creation and sharing of new challenges. Our experience suggests that teachers gained a sense of ownership of the material when actively engaged in adapting the scheme of work, which further helped build confidence.

In contrast, we also saw evidence of many teachers lacking confidence to deviate from the structure of the curriculum, allowing learning through iterative exploration. It may be that non-specialist teachers are not confident in how they could achieve learning outcomes without structure; or perhaps they find answering student questions challenging. We found the three teacher personas articulated in (Sentance et al. 2017) matched our observations: *Inspirers* created open-ended resources for students and shared with the wider community; *providers* created structured resources and focused on learning outcomes; *consumers* used existing resources and focused on engagement rather than learning. These provide a starting point to consider the design of teacher tools to support learning and assessment, encouraging teachers to develop from consumers to inspirers.

6.3 Create Inclusive Learning Experiences

The Torino Learning Environment was designed to enable inclusive learning experiences. We saw several examples in which teachers took this to heart, bringing multiple children with mixed visual abilities together to learn on a "level playing field," or supporting friendship between a blind and sighted child. Yet, many teachers rejected the idea of having children work together, seeing it as difficult and a practical annoyance. Despite aligning to an educational vision of inclusion, during the evaluation we had to respect existing practices of teaching non-accessible subjects one-to-one rather than inclusively in the classroom (Metatla, Serrano, et al. 2018). Cultural change takes time and cannot be achieved through the deployment of technology alone (Wing 2009). Technology can be designed, however, to encourage teacher evangelists to share their practices, a first step in transferring ownership of the technology from the research team to the teacher community for long-term viability (Taylor et al. 2013).

6.4 Limitations

Very few teachers actively commented on the design of Torino and its link to learning. While this is not surprising, as teachers are not researchers, it does make it hard to relate learning to aspects of the design. It would have been helpful to have had video data of students using Torino as done by (Lechelt et al. 2018; Wyeth 2008). However, this was impractical given the scale of this deployment. We had 24 local authorities (and even more schools) spread out across the UK. This was further moderated by constrained access to schools, since our relationship was with a peripatetic teacher rather than a school. Negotiating access or video opportunities with so many parties was entirely impractical and could have reduced willingness to participate in the research. We felt that relying on video analysis of a subset of opportunistically sampled students that we could get access to, given the extreme variation of age, visual, and cognitive abilities, may have led us to draw too heavily on data not representative of the larger cohort. As a result, what we can say about the relationship between learning and design is somewhat constrained.

6.5 Points of Reflection

Below are key points of reflection that may provide useful guidance to future studies of inclusive learning technologies:

- Consider whether a single learning endpoint is appropriate;
- Scaffold teacher's understanding of potential learning outcomes throughout appropriate frameworks;
- Provide strategies and technology for teachers to share student work with others for support;
- Invite and value the reporting of whole-child learning;
- Identify strategies that balance open-ended learning assignments and structured resources;
- Be prepared to support existing as well as new practices;
- Embed tools for community support to encourage culture shift and community ownership.

7 CONCLUSION

The technologies that we design play a mediating role in our interactions with the world (Verbeek 2015). If we are to achieve an inclusive learning culture, we need to provide tools that embed this philosophy in the design and help transform existing practices. Careful design can go beyond an engaging experience to support non-specialist teachers to co-produce learning and eventually take ownership of the technology.

8 ACKNOWLEDGEMENTS

We would like to acknowledge the teachers and learners who took part in the Beta Trial.

9 REFERENCES

Bach, Stephen, Ian Kessler, and Paul Heron. 2004. "Support Roles and Changing Jobs Boundaries in the Public Services : The Case of Teaching Assistants in British Primary Schools." In *Proceedings of the 22nd Annual International Labour Press Conference.*, 5–7.

Baker, Catherine M., 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 2015 CHI Conference on Human Factors in Computing Systems*, 3043–52.

Bouck, Emily C. 2016. "A National Snapshot of Assistive Technology for Students with Disabilities." *Journal of Special Education Technology* 31 (2): 4–13.

Braun, Virginia, and Victoria Clarke. 2006. "Using Thematic Analysis in Psychology." *Qualitative Research in Psychology* 3 (2): 77–101.

Brennan, Karen, and Mitchel Resnick. 2012. "New Frameworks for Studying and Assessing the Development of Computational Thinking." In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, 25.

Brule, Emeline, Gilles Bailly, Anke Brock, Frédéric Valentin, Grégoire Denis, and Christophe Jouffrais. 2016. "MapSense: Multi-Sensory Interactive Maps for Children Living with Visual Impairments." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 445–57.

Burg, Brian, Adrian Kuhn, and Chris Parnin. 2013. "1st International Workshop on Live Programming (LIVE 2013)." In *Proceeding of the 2013 ICSE International Conference on Software Engineering*, 1529–30.

Burgstahler, Sheryl E., and Richard E. Ladner. 2007. "Increasing the Participation of People with Disabilities in Computing Fields." *Computer* 40 (5): 94–97.

Cooper, Stephen, Lance C. Pérez, and Daphne Rainey. 2010. "K--12 Computational Learning." *Communications of the ACM* 53 (11): 27–29.

Cullen, Clare, and Oussama Metatla. 2018. "Multisensory Storytelling: A Co-Design Study with Children with Mixed Visual Abilities." In

Proceedings of the 2018 IDC Conference on Interaction Design and Children, 557–62.

DeLyser, Leigh Ann, Bryan Mascio, and Kelsey Finkel. 2016. "Introducing Student Assessments with Evidence of Validity for NYC's CS4All." In *Proceedings of the 2016 WiPSCE Workshop in Primary and Secondary Computing Education*, 17–26.

Department for Education. 2013. "Computing Programmes of Study : Key Stages 1 and 2 National Curriculum in England." http://www.computingatschool.org.uk/data/uploads/primary_national_curriculum_-_computing.pdf.

Duncan, Caitlin, Tim Bell, and Steve Tanimoto. 2014. "Should Your 8-Year-Old Learn Coding?" In *Proceedings of the 2014 WiPSCE Workshop on Primary and Secondary Computing Education.*, 60–69.

Eccles, Jacquelynne, and Allan Wigfield. 2002. "Motivational Beliefs, Values, and Goals." *Annual Review of Psychology* 53 (1): 109–32.

Giordano, Daniela, Francesco Maiorana, Andrew Paul Csizmadia, Simon Marsden, Charles Riedesel, Shitanshu Mishra, and Lina Vinikienė. 2015. "New Horizons in the Assessment of Computer Science at School and beyond: Leveraging on the Viva Platform." In *Proceedings of the 2015 ITiCSE on Working Group Reports*, 117–47.

Gray, Colette. 2009. "A Qualitatively Different Experience: Mainstreaming Pupils with a Visual Impairment in Northern Ireland." *Euro. J. Special Needs Education* 24 (2): 169–182.

Grover, Shuchi, Stephen Cooper, and Roy Pea. 2014. "Assessing Computational Learning in K-12." In *In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 57–62.

Hadwen-Bennett, Alex, 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).

Horn, Michael S., and Robert J. K. Jacob. 2007a. "Designing Tangible Programming Languages for Classroom Use." In *Proceedings of the 2007 TEI Conference on Tangible and Embedded Interaction*, 159–162. New York, NY, USA: ACM.

———. 2007b. "Tangible Programming in the Classroom with Tern." In *Proceedings of the 2007 Conference on Human Factors in Computing Systems Extended Abstracts*, 1965–1970. New York, NY, USA: ACM.

Horn, Michael S., Erin Treacy Solovey, and Robert J. K. Jacob. 2008. "Tangible Programming and Informal Science Learning." In *Proceedings of the 2008 IDC Conference on Interaction Design and Children*, 194.

Horn, Michael S, Erin Treacy Solovey, R Jordan Crouser, and Robert J K Jacob. 2009. "Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education." *Proceedings of the 2009 CHI Conference on Human Factors in Computing Systems.* 32: 975.

Israel, Maya, Quentin M. Wherfel, Jamie Pearson, Saadeddine Shehab, and Tanya Tapia. 2015. "Empowering K–12 Students with Disabilities to Learn Computational Thinking and Computer Programming." *TEACHING Exceptional Children* 48 (1): 45–53.

Kallia, Maria. 2017. "Assessment in Computer Science Courses: A Literature Review." https://royalsociety.org/~/media/policy/projects/

Kane, Shaun K, Varsha Koushik, and Annika Muehlbradt. 2018. "Bonk: Accessible Programming for Accessible Audio Games." *In Proceedings of the 2018 IDC Conference on Interaction Design and Children*, 132–42.

Kasanen, Kati, Hannu Raty, and Anna-Leena Eklund. 2009. "Elementary School Pupils' Evaluations of the Malleability of Their Academic Abilities." *Educational Research* 51 (1): 27–38.

Lechelt, Zuzanna, Yvonne Rogers, Nicolai Marquardt, and Venus Shum. 2016. "ConnectUs: A New Toolkit for Teaching about the Internet of Things." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 3711–14.

Lechelt, Zuzanna, Yvonne Rogers, Nicola Yuill, Lena Nagl, Grazia Ragone, and Nicolai Marquardt. 2018. "Inclusive Computing in Special Needs Classrooms: Designing for All." In *In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 517.

Lee, Irene, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. "Computational Thinking for Youth in Practice." *ACM Inroads* 2 (1): 32.

Leuders, J. 2016. "Tactile and Acoustic Teaching Material in Inclusive Mathematics Classrooms." *British Journal of Visual Impairment* 34 (1): 42–53.

Lister, Raymond, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, et al. 2004. "A Multi-National Study of Reading and Tracing Skills in Novice Programmers." *ACM SIGCSE Bulletin* 36 (4). ACM: 119–50.

Ludi, Stephanie, Lindsey Ellis, and Scott Jordan. 2014. "An Accessible Robotics Programming Environment for Visually Impaired Users." *Proceedings of the 2014 ASSETS International Conference on Computers & Accessibility*, 237–38.

Maloney, John, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. "The Scratch Programming Language and Environment." *Trans. Comput. Educ.* 10 (4): 16:1–16:15.

Martinez, Cecilia, Marcos J. Gomez, and Luciana Benotti. 2015. "A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform." In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 159–64.

Metatla, Oussama, and Clare Cullen. 2018. ""Bursting the Assistance Bubble": Designing Inclusive Technology with Children with Mixed Visual Abilities." In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 346.

Metatla, Oussama, Marcos Serrano, Christophe Jouffrais, Anja Thieme, Shaun Kane, Stacy Branham, Émeline Brulé, and Cynthia L. Bennett. 2018. "Inclusive Education Technologies: Emerging Opportunities for People with Visual Impairments." In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, W13.

Metatla, Oussama, Anja Thieme, Emeline Brulé, Cynthia Bennett, Marcos Serrano, and Christophe Jouffrais. 2018. "Toward Classroom Experiences Inclusive of Students with Disabilities." *Interactions* 26 (1): 40–45.

Milne, Lauren R, and Richard E Ladner. 2018. “Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments.” In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 69.

Moon, Nathan W., Robert L. Todd, David L. Morton, and Emily Ivey. 2012. “Accommodating Students with Disabilities in Science, Technology, Engineering, and Mathematics (STEM): Findings from Research and Practice for Middle Grades through University Education.” http://www.catea.gatech.edu/scitrain/accommodating.pdf.

Morrison, Cecily, Nicolas Villar, A Thieme, Zahra Ashktorab, Eloise Taysom, Oscar Salandin, Daniel Cletheroe, et al. 2018. “Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities.” *Human Computer Interaction*. https://doi.org/10.1080/07370024.2018.1512413

Peyton-Jones, Simon. 2018. “Project Quantum: Tests Worth Teaching To.” https://community.computingatschool.org.uk/resources/4382/single

Peyton-Jones, Simon, Simon Humphreys, and Bill Mitchell. 2013. “Computing at School in the UK : From Guerrilla to Gorilla.” https://www.microsoft.com/en-us/research/publication/computing-at-school-in-the-uk-from-guerrilla-to-gorilla/.

RNIB. 2016. “Children and Young People England RNIB Evidence-Based Review.” https://www.rnib.org.uk/sites/default/files/Eviden

Rogers, Yvonne, Venus Shum, Nic 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): 74–77.

Román-González, Marcos, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández. 2017. “Which Cognitive Abilities Underlie Computational Thinking? Criterion Validity of the Computational Thinking Test.” *Computers in Human Behavior* 72: 678–91.

Sentance, Sue, and Andrew Csizmadia. 2016. “Computing in the Curriculum: Challenges and Strategies from a Teacher’s Perspective.” *Education and Information Technologies*, 1–27.

Sentance, Sue, and Jane Waite. 2018. “Computing in the Classroom: Tales from the Chalkface.” *It-Information Technology* 60 (2): 103–12.

Sentance, Sue, Jane Waite, Lucy Yeomans, and Emily MacLeod. 2017. “Teaching with Physical Computing Devices: The BBC Micro: Bit Initiative.” In *Proceedings of the 2017 WiPSCE Workshop on Primary and Secondary Computing Education*, 87–96.

Sorva, Juha. 2013. “Notional Machines and Introductory Programming Education.” *ACM Transactions of Computing Education* 13 (2).

Stefik, Andreas M., 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 2011 CSE Technical Symposium on Computer Science Education*, 571.

Sullivan, Amanda, Mollie Elkin, and Marina Umaschi Bers. 2015. “KIBO Robot Demo: Engaging Young Children in Programming and Engineering.” In *Proceedings of the 2015 IDC International Conference on Interaction Design and Children*, 418–21.

Tanimoto, S. 1990. “VIVA: A Visual Language for Image Processing.” *J. Vis. Languages Computing*, 127–39.

Taylor, Nick, Keith Cheverst, Peter Wright, and Patrick Olivier. 2013. “Leaving the Wild: Lessons from Community Technology Handovers.” In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems*.

Thieme, Anja, Cecily Morrison, Nicolas Villar, Martin Grayson, and Siân Lindley. 2017. “Enabling Collaboration in Learning Computer Programing Inclusive of Children with Vision Impairments.” In *Proceedings of the 2017 DIS Conference on Designing Interactive Systems*, 739–52.

UK Parliment. 1981. “The 1981 Education Act.” *Http://www.legislation.gov.uk/Ukpga/1981/60/Enacted*.

Utting, Ian, Stephen Cooper, Michael Kölling, John Maloney, and Mitchel Resnick. 2010. “Alice, Greenfoot, and Scratch – A Discussion.” *Trans. Comput. Educ.* 10 (4): 17:1–17:11.

Verbeek, Peter-Paul. 2015. “Beyond Interaction: A Short Introduction to Mediation Theory.” *Interactions* 22 (3): 26–31.

Virnes, Marjo, Erkki Sutinen, and Eija Kärnä-Lin. 2008. “How Children’s Individual Needs Challenge the Design of Educational Robotics.” In *Proceedings of the 2008 ICD International Conference on Interaction Design and Children*, 274–281.

Waite, Jane, Paul Curzon, William Marsh, Sue Sentance, and Alex Hawden-Bennett. 2018. “Abstraction in Action: K-5 Teachers’ Uses of Levels of Abstraction, Particularly the Design Level, in Teaching Programming.” *International Journal Of Computer Science Education In Schools*.

Wing, Michael. 2009. “On Technology and Cultural Change.” *SIGSOFT Softw. Eng. Notes* 34 (2): 4.

Wyeth, Peta. 2008. “How Young Children Learn to Program with Sensor, Action, and Logic Blocks.” *The Journal of the Learning Sciences* 17 (4): 517–50.

Zaman, Bieke, Vero Vanden Abeele, Panos Markopoulos, and Paul Marshall. 2012. “The Evolving Field of Tangible Interaction for Children: The Challenge of Empirical Validation.” *Personal and Ubiquitous Computing* 16 (4): 367–78.

Zhou, Li, Amy T. Parker, Derrick W. Smith, and Nora Griffin-Shirley. 2011. “Assistive Technology for Students with Visual Impairments: Challenges and Needs in Teachers’ Preparation Programs and Practice.” *Journal of Visual Impairment & Blindness* 105 (4): 197–210.

Zuckerman, Oren, Tina Grotzer, and Kelly Leahy. 2006. “Flow Blocks as a Conceptual Bridge between Understanding the Structure and Behavior of a Complex Causal System.” In *Proceedings of the 2006 International Conference on Learning Sciences*, 880–86. International Society of the Learning Sciences.