Physical programming inclusive of young children with visual disabilities
Click the green flag. Move the mouse to eat the small fish.
Inside ‘If’ Statement. Block 4 of 6 in workspace.


Challenges

1. Assistive technology proficiency
2. Significant memory demand
3. Lack conceptual cognitive structures
Project Torino
Torino Design Journey
Design Principals
1. Persistent Program Overview
2. Liveness
3. Low Floor, High Ceiling

Sequence

Complex program
4. Works across Visual Abilities
5. Enables progression
Torino Beta Study
Torino Beta

75 Children
30 Teachers
24 Local Authorities
Deployment

Measuring Computational Learning

**Method Considerations**

1) Students are diverse in their abilities;

2) Teachers are non-specialists;

3) Data collection by researchers is restricted due to logistics.

**Method Approach**

1) Validated questionnaire to measure engagement;

2) Measurement of motivational construct;

3) Teacher reported learning outcomes.
What do teacher’s think?

<table>
<thead>
<tr>
<th>Statement</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>I think Torino is a good tool for teaching coding to visually impaired children.</td>
<td>4.9(5)</td>
</tr>
<tr>
<td>I found some of the computing concepts hard to understand.</td>
<td>2.4(1)</td>
</tr>
<tr>
<td>Teaching with Torino helped me to improve my own computing subject knowledge.</td>
<td>4.2(5)</td>
</tr>
<tr>
<td>The teachers’ guide was hard to follow.</td>
<td>1.7(1)</td>
</tr>
<tr>
<td>I would like to use Torino to teach coding in the future.</td>
<td>4.8(5)</td>
</tr>
</tbody>
</table>

1 = disagree strongly; 5 = agree strongly

100% of teachers strongly agreed or agreed that they would like to use Torino to teach coding in future.
Engagement

How excited do you get before a Torino session?

4.32/5
Motivation

How many stars would you like to give yourself now for your coding / programming ability?

Pre

Post

$R = -0.73$
1. Persistent Program Overview

“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)”
2. Liveness

“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)
3. Low Floor, High Ceiling

“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)
4. Works across Visual Abilities

“The pupils have worked extremely well together and have helped each other to rapidly pick up many of the concepts and vocabulary used. Here is a video of successful joint working and happy collaboration!”
Teacher Reported Learning

Use of Correct Vocabulary
When completing activities, the children now often use correct key terms – ‘sequence’, ‘thread’, ‘parameter’ etc. (T10)

Problem-Solving
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)

Inclusive Education
The students, all with a visual impairment from different schools, learned how to work together as a team.
Questions for Discussion

1. How might a physical programming language help young children without a visual impairment who are struggling to learn using a block-based language?

2. What can we learn from the success of students physically following their code?

3. How should we measure success of a new programming language?

4. Physical programming can be exciting, but it's more expensive than software. How can we enable such opportunities?