Inclusive Computational Thinking and Collaborative Learning using an Accessible Remote Robotic Kit

Diana Sofia Matos Mendes

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Hugo Miguel Aleixo Albuquerque Nicolau

Examination Committee

Chairperson: Prof. Pedro Tiago Gonçalves Monteiro
Supervisor: Prof. Hugo Miguel Aleixo Albuquerque Nicolau
Members of the Committee: Prof. Ana Cristina de Oliveira Tomé Lopes Pires

October 2021
Acknowledgments

I would like to thank my parents and my two brothers for their unconditional support, friendship, encouragement and caring over all these years.

Also, to my grandparents and Sr.Paredes, who cared for me since I was little and taught me the importance of connecting with one another and Mother Nature. I would like to thank Átila for reminding me to live in the present, be calm and assertive, and strive to be better every day.

I want to thank Joel for his love and caring, for never letting me give up and for taking me on the most random and crazy adventures.

To my friends for all of their support over these years. Especial thanks to Peter, Mafalda, Nuno and Inês. Thank you “Team Convocatória”, “Team Potato” and “Heavenly Corner” for long night conversations, gaming sessions and going outs.

To Sofia for helping me during this entire journey.

Finally, I would also like to acknowledge my dissertation supervisors Prof. Hugo Nicolau and Prof. Isabel Neto. for their insight, support and sharing of knowledge that has made this Thesis possible. To all the professors and investigators who helped me in this journey. A special thanks to Tiago Guerreiro, Ana Pires and Francisco Teotónio for sharing their ideas with me. To Filipa Rocha for her willingness to always help, especially conducting the user studies, and for sharing her previous work with me. To Gonçalo, for inspiring the creation of our system. To Lúcia Abreu, who made the first version of the floor caps. And finally to Cristiana Antunes for sharing the Ozobot Evo library with me.

To each and every one of you – Thank you.
Abstract

Computational Thinking is a fundamental skill to be developed from an early age, and it can be fostered through the use of robotic coding kits. To make them accessible for groups with children with mixed visual abilities, the coding kits usually have tangible blocks (input) that program a robot (output) and use audible cues and feedback. When children collaborate in these activities they (also) feel more integrated and if the system promotes collaboration it is also promoting inclusion within the children.

Considering these kits are accessible and engaging, it is naturally nurtured. However, it can be forced through competitive activities between groups or the attribution of roles that separate the information/functions between two or more children. Nevertheless, existing solutions to foster CT and collaboration require children's presence and do not allow them to carry out the activities remotely. Moreover, this co-located requirement places children in non-balanced situations about the system’s knowledge. All children have access to tangible and audible information, but only sighted children can understand visual information. Therefore, we propose a coding kit that uses blocks to code a robot (Ozobot Evo) and allows two children to play Sokoban remotely. Our system separates the input and the output while assigning different responsibilities to each child (they do not share working areas). So, they need to collaborate to reach their common goal of completing the different tangible Sokoban levels.

Keywords

Visual Impaired; Children; Robots; Coding Kit; Collaboration; Computational Thinking; Accessible System; Remote System; Sokoban.
Resumo

O Pensamento Computacional é uma competência fundamental que deve ser desenvolvida durante a infância, podendo ser promovida com recurso ao uso de kits de programação robóticos. Assim sendo, de modo a torná-los acessíveis para grupos com crianças com deficiências visuais, normalmente estes kits são providos de blocos tangíveis com dicas tácteis (input) e de feedback auditivo, que programam um robot (output). Também a Colaboração entre crianças deve ser promovida, por ser essencial para que se sintam integradas. Pelo facto de serem acessíveis e envolventes, os kits de programação acabam por nutri-la naturalmente. No entanto, podemos forcar a Colaboração entre crianças por meio de atividades competitivas entre grupos ou através da atribuição de papéis que separam as informações/funções do sistema entre elas. No entanto, as soluções atualmente existentes para promover o Pensamento Computacional e a Colaboração exigem a presença física das crianças, não permitindo realização das atividades à distância. Adicionalmente, este requisito co-localizado coloca-as em situações desequilibradas em relação à omnisciência do sistema, pois todas as crianças têm acesso a informações tangíveis e auditíveis, mas apenas crianças com visão poderão entender as informações visuais. Assim sendo, propomos um kit de programação que utiliza blocos tangíveis para codificar um robot (Ozobot Evo) e que permite que duas crianças joguem Sokoban remotamente. Como o nosso sistema separa o input e o output e atribui papéis a cada uma delas (restringindo-as às suas áreas de trabalho), elas terão inevitavelmente de comunicar entre elas para alcançarem seus objetivos de completarem os níveis tangíveis de Sokoban.

Palavras Chave

Crianças com Deficiência Visual; Robots; Kits de Programação; Colaboração; Pensamento Computacional; Sistemas Acessível; Sistema Remoto; Sokoban
Contents

1 Introduction 1
   1.1 Motivation ................................................. 3
   1.2 Previous Work ............................................. 4
   1.3 Problem .................................................. 4
   1.4 Approach ................................................ 5
   1.5 Goals ..................................................... 6
   1.6 Document Structure ...................................... 6

2 Related Work 7
   2.1 Computational Thinking for Children ........................ 9
   2.2 Accessible Programming Environments ....................... 13
   2.3 Collaborative Learning ................................... 19
   2.4 Discussion ............................................... 22

3 Accessible Remote Robotic Kit - Designing Process 25
   3.1 Game Designing .......................................... 27
      3.1.1 Formative User Study ................................. 28
      3.1.2 Findings and Discussion .............................. 28
      3.1.3 Conclusions .......................................... 29
   3.2 Sokoban Map .............................................. 30
      3.2.1 Ozobot Pushers ....................................... 30
      3.2.2 Crates ................................................ 31
      3.2.3 Map: Walls and Floor ................................. 31
      3.2.4 Map Levels ........................................... 32
   3.3 Instruction Blocks ........................................ 33
      3.3.1 Colour Preference Questionnaire ..................... 34

4 Architecture 37
   4.1 Web Application .......................................... 39
      4.1.1 Block Commander ..................................... 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2</td>
<td>Map Explorer</td>
<td>40</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Presential User</td>
<td>40</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Implementation Details</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Block Commander</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Magic Box</td>
<td>42</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Plate and Blocks</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Map Explorer</td>
<td>44</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Robot Movement and Lights</td>
<td>44</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Maps</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Keyboard Cover and Screen Reader</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>User Study</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>Research Question</td>
<td>49</td>
</tr>
<tr>
<td>5.2</td>
<td>Participants</td>
<td>49</td>
</tr>
<tr>
<td>5.3</td>
<td>Robotic Kit</td>
<td>50</td>
</tr>
<tr>
<td>5.4</td>
<td>Session Procedure</td>
<td>51</td>
</tr>
<tr>
<td>5.5</td>
<td>Data Collection and Analysis</td>
<td>52</td>
</tr>
<tr>
<td>5.6</td>
<td>Findings</td>
<td>53</td>
</tr>
<tr>
<td>5.6.1</td>
<td>General Findings</td>
<td>53</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Engagement</td>
<td>54</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Computational Thinking</td>
<td>55</td>
</tr>
<tr>
<td>5.6.4</td>
<td>Collaboration</td>
<td>56</td>
</tr>
<tr>
<td>5.7</td>
<td>Discussion</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>59</td>
</tr>
<tr>
<td>6.1</td>
<td>Achievements</td>
<td>61</td>
</tr>
<tr>
<td>6.2</td>
<td>System Limitations</td>
<td>61</td>
</tr>
<tr>
<td>6.3</td>
<td>Future Work</td>
<td>61</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>A</td>
<td>Informed Consent Form 1</td>
<td>69</td>
</tr>
<tr>
<td>B</td>
<td>Informed Consent Form 2</td>
<td>73</td>
</tr>
<tr>
<td>C</td>
<td>User Study</td>
<td>79</td>
</tr>
<tr>
<td>D</td>
<td>Questionnaire</td>
<td>91</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 The figure shows Ozobots during the experience of [1]. On the left the robots are customized with crafting materials; on the right they are used in a gaming activity.  

3.1 The image shows the two games’ second level - Maze on the top, and Sokoban at the bottom. For both cases, on the left, we have the sight of the Map Explorer and on the right is the Block Commander’s work-view.  

3.2 The image shows the three different 3D Printed Ozobot Pushers in chronological order.  

3.3 The image shows the 3D printed Crates.  

3.4 Ozobots moving on top of caps with different dimensions. From the left to the right: 2x2 cap, 3x3 cap and 4x4 cap.  

3.5 The image shows the design of different floor caps.  

3.6 Sokoban’s first level solution.  

3.7 The image shows respectively the second and third levels of Sokoban, from the first study we made. The second level needs at least two iterations, with a total of thirteen instructions given to the character. The third level needs at least six iterations, with a total of thirty four instructions given to the character.  

3.8 The image shows our design for the second and third levels of Sokoban.  

3.9 Plate and Blocks’ Designs.  

4.1 General Architecture of our remote robotic kit. Block Commander on the left and Map explorer on the right.  

4.2 Web Application Menu. With the button on the left, the child chooses to be the Block Commander. The button on the right turns the child into the Map Explorer.  

4.3 Communication between the Block Commander client and the Heroku Server.  

4.4 Communication between the Heroku Server and the Map Explorer Client.  

4.5 Interior UV Mapping of the Magic Box.
4.6 Circuit Diagram. The LED is connected to a DC-DC Step-Down Module Converter, that connects to the laptop through USB.

4.7 Magic Box Final Design. Slight from the top

4.8 Plate and Blocks’ Final Design.

4.9 Maps Final Designs.

4.10 Keyboard Cover.

5.1 Children’s Ages - Mean and Standard Deviation.

5.2 Sokoban Levels for Presential and Remote Scenarios.

5.3 Charts showing duration and the number of errors committed by children on each level, for both scenarios (remote and presential).

5.4 Chart showing children’s opinion about the inclusiveness of our system.

5.5 Charts showing children’s answers about the activities: engagement and difficulties.

5.6 Chart showing children’s responses about how much they enjoyed the activities, for both scenarios.

5.7 Charts showing children’s preference about the Roles Map Explorer and Block Commander.

5.8 Chart showing children’s opinion about how collaborative the activities were.

5.9 Chart showing how relevant children’s collaboration were, in their opinion.
List of Tables

2.1 Summary of the analyzed studies about CT, Collaboration and accessible Systems. . . . 22
5.1 Summary of Participants’ Demographic and Visual Information. . . . . . . . . . . . 49
5.2 Summary of the Participants’ background about programming and robots. . . . . . 50
5.3 Maps assignment to each pair of children . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
Acronyms

VI Visually Impaired
CT Computational Thinking
SNE Special Needs Educator
Introduction

Contents

1.1 Motivation ................................................................. 3
1.2 Previous Work ........................................................... 4
1.3 Problem ................................................................. 4
1.4 Approach ............................................................... 5
1.5 Goals ................................................................. 6
1.6 Document Structure ....................................................... 6
Robots have been growing in the domains of child development and education, becoming increasingly used tools for teaching Computational Thinking (CT) and enabling good levels of interaction and engaging experiences [2, 3].

As a matter of fact, they are being incorporated into coding kits to foster collaboration between children and the development of CT. However, to integrate Visually Impaired (VI) children into their proposed activities, these kits have become more accessible: they no longer rely solely on visual information. Instead, they integrate tangible and audible information to be perceived by every child [1,4–7]. As a result, they enable socialization and collaboration [3, 8, 9].

However, despite these activities and coding kits being accessible, since all information regarding the three senses is available to all children, they become unbalanced. Although sighted children have access to tangible and audible cues, children with VI have less visual information. Besides, these systems require children’s physical presence to be used.

1.1 Motivation

Computational Thinking has been standing out for the last few years due its level of importance, since it helps people think and explore alternative ways to solve problems and live more autonomous lives.

In fact, it is a skill that uses computer science concepts, such as abstraction, algorithms, automation, decomposition, debugging and generalization, to develop analytical thinking (similar to mathematics), engineering and scientific thinking. It enables students to learn how to translate a problem into an expressive process, declaring ideas in new ways, subdividing them into simple steps and solving each step at each time. [2, 10–13]. Although its nature addresses tasks directly linked with programming exercises (such as sequencing, debug, problem decomposition, e.g.), CT is more comprehensive than that and can be seen as an "expressive process that allows for new ways to communicate ideas" [10,14]. Therefore, researchers have been highlighting its value to stimulate students from early ages, so that they can become independent adults [14].

Moreover, if children collaborate while solving activities that foster CT, they can reach higher levels of productivity [2]. In fact, collaboration foments caring, supportive and committed relationships, aid psychological health, self-esteem, and social competence to children, and reduces the propensity of children with VI to isolation [7, 15]. Thus, when designing coding kits for children with VI, we must not forget to incorporate collaboration. It can be fostered when fomenting competition between groups, for instance [6]. However, if activities that were initially thought to develop CT are accessible and engaging for all participants, collaboration and communication between children would naturally arise.

Therefore, the development of CT and collaboration must not require children's physical presence. If it was possible to conceive it in a remote context, children could collaborate and solve CT activities
whenever they want and anywhere.

1.2 Previous Work

Previous work has started to look into accessible programming environments and inclusive systems to foster CT and collaboration between children with mixed visual abilities [1, 6, 10, 13, 16].

Since tactile differentiation's and sounds are more appropriate for VI children (they depend more on touch, audio feedback, instruments and their peers) than visual cues, the addressed solutions used coding kits composed of programmable robots with tangible elements [17]. However, previous work also highlighted the importance of not neglecting appealing visual elements, so that sighted children would feel more engaged in completing the activities and collaborating with VI children. Overall, as a result of the experiences with children, they had fun using the systems while collaborating during CT activities [1].

However, more work in this area is still needed. During these activities, the kits were unbalanced for VI children due to all of the system's information being available for everyone. Despite all of them having access to tangible and audible information, completely blind children could not perceive visual information, while their sighted peers also understood visual cues. Therefore, a system capable of dividing children's domains of omniscience and placing them at the same level, (to our knowledge) remains to be done.

Finally, all of the addressed activities were co-located, requiring children's physical presence. Due to the pandemic, this way of operating the system has become somewhat outdated. Besides, children should develop these skills (CT and collaboration) whenever they want and not be limited to school activities (for instance).

1.3 Problem

Visually Impaired children would benefit from the promotion of CT as they can sense more challenges due to their less autonomy, spatial awareness, and visual perception constraints. Therefore, CT activities could positively impact them by enriching their spatial awareness. Plus, by having similar experiences and equal access to learning activities, they could live more autonomously and follow equal opportunities (as their sighted peers) in their professional careers [18, 19].

Despite the presented assistive solutions for VI children, to our knowledge, all of them were limited to co-located activities. The only attempt where children were not together was in the experience [17], but that relied on an analysis of the importance of a robot being a mediation tool while working with children.

Therefore, children with and without visual impairments should have the opportunity of working together in remote, inclusive and educational environments that foster collaboration. As a result, these
kits would balance the system’s knowledge between children and make activities engaging for them.

However, to our knowledge, systems that propose to carry out these remote activities - allowing children to join whenever they want for collaborative and CT applications -, do not exist yet. Most of them depend on school use, which is a constraint for children to develop those skills at home, during leisure activities or online classes.

1.4 Approach

We have pursued a solution that incorporated robots into an accessible kit - with visual, audible and tangible information -, since robots have been proven to help fostering collaboration and teaching CT/navigation notion to mixed visual abilities children [20].

To ensure children would collaborate using our kit, we have decided to create an interdependent activity. Assigning roles that would attribute specific system’s functions to each child. While one child will be responsible for coding the robot by assembling tangible coding blocks, the other will observe the robot’s movements on a map and discuss with his peer its following actions.

However, in co-located activities, sighted children have more visual access to the activity therefore they tend to command and dominate system and activity interactions [21]. Additionally, they may interfere with their VI peers’ working spaces. Thus, assigning roles may be not enough to ensure this division of functions and ensure collaboration between peers, as the access to each others working spaces also means the system becomes unbalanced for VI children, who have more limited access to visual information than his sighted peer.

Thus, by making a remote kit we will control what children know about the system. In our study each child only knows about his working space and the non-visual information exchange with his peer, allowing a balanced access to the systems’ and activity information between them.

Finally we design a learning through play activity, in line with existing systems that used to carry out non-school-related activities, while still teaching CT key concepts to children [2, 18, 22], we decided to design a system to be a game. Our choice of Sokoban game is later explained in chapter 3.

In order to understand if the kit responds to our requirements, we conducted a user study where we gave two kits to each pair of children, to evaluate our three hypothesis: H1) The remote robotic kit can foster collaboration and CT; H2) The remote robotic kit can be engaging; H3) Our remote robotic kit was as much engaging and support effective CT learning and peers collaboration as its co-located variant.
1.5 Goals

With this dissertation, we intend to provide an Accessible Remote Robotic Kit that promotes collaboration between children with mixed visual abilities and that fosters their CT development. We aim to meet the following goals:

- Design small and engaging tangible blocks for coding a robot;
- Choose an engaging game for children (that would provide CT learning) and transform it into a tangible activity;
- Perform user tests with mixed visual abilities children to validate our solution;
- Understand if the kit fostered collaboration between children and supported CT learning;
- Understand if the kit was capable of balancing the systems’ information between children;
- Understand the differences between our remote kit and a co-located variant.

1.6 Document Structure

Chapter 2 highlights the importance of fostering Computational Thinking and collaboration between children with mixed visual abilities at early ages. Besides, it also emphasizes that the existing systems are unbalanced regarding the division of their information among children. Finally, it shows that all systems require their physical presence, impeding VI children from developing these skills with their friends under engaging activities in remote contexts.

Chapter 3 focuses on the designing process of our remote coding kit. It explains our design choices, namely the integration of the Sokoban game into our System.

Chapter 4 describes all the tools we have used to implement our final solution. It also presents our final robotic Coding Kit.

Chapter 5 takes into account the user studies - analysis and discussion of the results.

Finally, chapter 6 takes a grant of the conclusions.
2.1 Computational Thinking for Children ................................. 9
2.2 Accessible Programming Environments ............................ 13
2.3 Collaborative Learning .................................................. 19
2.4 Discussion ............................................................. 22
Vision allows the construction of cognitive maps which can rely on two different representations: allocentric or egocentric. VI children can also build cognitive maps. However, they rely on other modalities such as movement, touch, and audition, e.g. [17].

The lack of vision makes tasks like locating or identifying objects complicated (especially when others manipulate them) [4,8]. Moreover, most times when children start an activity, they do not share the same referential (for instance, VI people tend to use egocentric representations more often than allocentric [17]), neither the same vocabulary, which difficults collaboration. Plus, it can be accentuated when the activities are not accessible for VI children or when its elements are simplified and not engaging for their sighted peers.

Therefore, when VI children are integrated into mainstream schools, they feel different from their peers and excluded from their groups, and this foments their isolation [7, 15, 23]. However, with the transition to a remote education context, adapting to the school dynamics is starting to be overcome. Plus, both children lose the perception of the other’s visual reality, which allows them to be on almost similar visual perception levels, making them feel and interact as equals.

Thus, collaboration is important. Besides fomenting important social skills and notions, children can learn new concepts through their interaction with others and develop new functional skills [24]. However, to collaborate, both children have to understand the constituents of activities to feel on the same level of equality. So, audio and haptic feedback can inform about other persons’ actions and the system’s state [4, 18]. Finally, children prefer group interactions with robots, and they are a standard tool in learning, coaching, and therapy [8, 25]. As cognitive development can influence learning how to program robots [12], children’s age is a relevant factor in setting collaborative activities.

The following sections present some of valuable work and experiences related to computational thinking, accessibility and collaboration. In every section we will address its integration for VI children.

### 2.1 Computational Thinking for Children

Computational Thinking is a fundamental skill to be developed, since it supports people to solve problems independently, helping them become autonomous adults. Moreover, when it starts to be introduced in early ages, it can also be promoting STEM - Science, Technology, Engineering and Mathematics [10, 11]. In fact, CT is defined as a skill to solve problems through algorithms and creativity, by designing systems and developing a sense of technology fluency. Once it uses "concepts of computer science to formulate and solve problems", it shares similarities with scientific, mathematical and engineering thinking [2, 10–12]. Some of the CT processes may include abstraction, algorithm and procedures, debugging, data collection, problem decomposition and pattern recognition [22].

Given the relevance of this capability, there has been previous research showing that the integration
of robotics programming exercises can lead to its development [14].

Also, the programming level should take into consideration children’s ages, since first and second graders present different levels of mastery in the domain of learning how to program [11], and verbal reasoning plays an important role. In fact, first-grade children are developing observation, deducing causes-and-effect relations and using simple sequences of words when expressing thoughts; and second graders are more interested into understanding/developing theories and can use inference when communicating ideas. Thus, the mastery of certain processes (such as expressing reasoning verbally) can positively impact the children’s ability to perform logical operations [12].

Moreover, the range of ages of the children makes the comprehension about quantitative (sum, subtraction, division and multiplication commands, for example) and categorical (similar, different, etc.) skills irrelevant to this stage. The "shapers" of children’s programming processes that actually play direct roles in this phase are shortened to sequencing and cause-effect relations [11]. The same logic applies when thinking about syntax errors, since missing a white space would only get children frustrated and despise the important part of the experience [14], that is the CT, navigation notion, e.g. In fact, the real point is that they understand sequences, loops, conditions and debugging [10].

Therefore, previous studies have been addressing this through robots that answer to children’s commands. Through coding - a new "alphabet" for them - they have been taught about CT [14]. By facing creation and problem-solving processes, young children can bridge technological and physical worlds and explore the development of cognitive skills and domain-specific content [12]. The approaches to promote CT with spatial activities can either include fully virtual environments (like Scratch, for example), virtual programming environments with tangible output (Blocky/Dash), tangible programming environment with virtual output (Cosmo, AWBIE) or fully tangible programming environments (Wonder Puzzlets for DASH, Clementoni’s DOC robot, Tactopi, etc.) [2,6,10,14,22].

However, when designing coding kits for children, they should not highly rely on "problem-solving" activities. Instead of emphasizing coding as a logic game, built upon solving puzzles activities, it should be understood as a means to an end, expressed by using metaphors to foster creative learning. For example, ScratchJr is an introductory programming language that enables children to explore computer programming concepts through creative expression, since it allows them to create their own digital objects, build animations and tell stories [11]. The same approach is taken within the experience with Kibo [10] - since it is based on the idea of "coding as a playground" which makes children more socially, physically, emotionally and creatively involved.

In the next paragraphs, we present some examples of studies that tried to address the design and development of activities promoting CT with children:

Bee-Bot [2]: a very colourful and easy-to-use robotic kit, made for young children. It is resistant and has a soft texture, with simple and friendly functionality. For moving the robot children had to
press the buttons on the top of Bee-bot - so it uses tangible inputs and outputs. During the study, a carpet/map where the robot was supposed to move was integrated into the kit. At the end of the experience, educators proposed to add a labyrinth to the carpet, to add a gamification factor to the activities.

**KIBO** [10]: this robot was used to integrate CT in the curriculum of children, by exploring organization, sequences and consequences concepts, as well as the composition of computing systems and how the different parts were linked with each other - the blocks (software) made the robot (hardware) move. In the experience children have been introduced to step-by-step activities to get familiarized with programming concepts and skills. Then specific objectives and instructions to program the robot were given. The main goals of teaching children fundamental CT concepts and coding skills were achieved once they understood "sequencing - ordering a sequence of steps to perform actions"; "repeats - performing the same sequence several times"; "conditionals - decisions related to events or actions"; and "debugging - finding and fixing errors in the code". Plus, were able to manipulate motors, platforms and sensors with KIBO, as well as integrate arts, crafts and recycled materials into their final robots projects. Also, during some of these experiments, teachers showed that CT could be integrated with other curriculum subjects. So, different challenges have been proposed (like curricular adaptation to work on literacy skills; or more abstract ones as music or dancing; games (making Kibo travelling different routes), e.g.) and they mostly ended up with children mastering KIBO and integrating it with social sciences. In fact, by testing their outcomes it was enhanced that their memory skills were better and they learnt how to sequence increasingly complex programs and master all of KIBO's syntax rules. Therefore, the result was really positive since it succeeded at teaching coding while engaging them into a playful appropriate learning experience. Thus, this experience highlights some effective techniques to teach CT through coding in classrooms with children.

**TACTOPI** [22]: children used tangible blocks to move a robot (a ship with sensors (for obstacle avoidance), lights and a buzzer) in a map, which was a very engaging inclusive activity that fostered spatial notion and CT. Because the map had 3D tangible elements with different textures, it was accessible to VI children (they could differentiate ocean from earth, for example) and allowed the development of mental models and spatial knowledge about new spaces/environments, empowering route wayfinding [26]. Plus, this kit had multi-sensory elements such as: audio narrative, sound cues, audible warnings, sounds of the sea, wind and animals, graspable elements of navigation, collectable coins e.g. which really engaged children in an immersive way. Plus, soundscapes and audio-books promote creativity and imaginative thinking [15]. This playful environment's goal was to use the ship, remotely controlled by a helm, to complete levels while moving it on the map. Therefore, children planned strategies and created sequences of instructions and navigating the ship made them use mental representations of their own bodies and the environment, which supported CT activities.
DASH [6]: Tangibles are recognized to reduce cognitive demand, once thinking is linked with motor activities. The experience with DASH used a tangible robot-based programming environment that enhanced the development of CT (while coding the robot) and spatial skills (like orientation, navigation and spatial cognition). In fact, educators saw tangible components and cues as a reinforcement for children's perceptual abilities; and the robot as a motivational artefact. Furthermore, maps allowed children to explore the space and bounds of the workspace and promoted orientation skills. Educators suggested them to represent a real location (a residential area), and they recommended to explicitly define the distance of each step of the robot (each step unit was equivalent to one EVA foam tile), to make it easier for children to develop mental models of how the blocks affect the robot. In fact, the programming knowledge was identified by two key indicators: the ability to match a programming command with its outcome - all children except one understood this - and the ability to create a program using the right commands in the right order - only older children were capable of doing it. Plus, the Play Block of the experience was not well perceived (children did not understand that the robot would execute the whole sequence attached to the Play Block once the button was pressed). Moreover, we could take a curious fact from this experience: although most children have first explored the robot by touch, depending on their age they made differently through that process - younger children were more spontaneous and abstractly examining the robot, while older ones were investigating it more consciously, sighting to understand its morphology and functionality.

Hungry Cat [18]: is a virtual game playable on a tablet. By controlling the Hungry Cat avatar, children gained spatial information about the environment. In fact, navigation - “obtaining spatial information from the environment and forming a spatial map about it” -, gathering spatial information about an area, mentally drawing a spatial layout of that area and being capable of localizing ourselves in the areas, are critical skills. This experience has shown that the children that took more steps to explore the environment have discovered more about each tile of the virtual room they were in. In fact, either young participants or with learning difficulties were able to create the spatial mental map of the locations of furniture in the rooms. But those who have effectively found the food and were successful at forming the spatial mental maps of the virtual rooms were those who explored more.

Torino [4]: is composed of physical components and a hub. It has assisted children while learning programming and CT concepts, regardless of their level of vision. Like other examples already mentioned, Torino did not require the learning of new language syntax, to avoid beginners’ frustrations. Instead, it used its physical components to be programmed, which helped children build up an understanding of the program’s structure, since VI children physically explored, hold and locally manipulate beads to build a ‘picture’ of things by actively touching them and listening to the sounds that bounce off objects. In fact, well-defined, easy-to-locate and consistent references points are often recommended when working with children with visual impairments, so they were encouraged to touch beads along the
program execution. This tracing and physically following the program had advantages in what concerns understanding better the mapping between bead container and audio representations, which helped children for debugging.

**Ozobots** [1]: are small robots, with two wheels and a colour sensor, designed to engage all children and they have already been successfully used on systems for VI people. Children controlled their movements by using coloured line drawings (and the robot followed it) or using a visual programming language. During the experience with Ozobots, children learn about their functionality by contacting them and their technologies. To introduce CT, they were challenged to create and impersonate their own robot characters. Then, they worked in groups to complete tasks, by receiving and executing instructions, as they were robots themselves. Because they played with robots first, they were conscious about breaking down a complex task into basic tasks and sequencing them. Then, they designed a map with Lego, tactile, scented materials and tape (for the routes). In general, children liked the game’s navigational component and were effective at generating and executing instructions and traversing the course. When designing coding kits and doing workshops with children, embedding learning objectives is essential. In fact, playing engaged them in content-based learning - like learning about a subject matter (space, for example) - and with technical aspects of the gaming material. Also, children felt more engaged in the activities when they used crafting customization, and designed physical courses and maps.

![Figure 2.1: The figure shows Ozobots during the experience of [1]. On the left the robots are customized with crafting materials; on the right they are used in a gaming activity.](image)

### 2.2 Accessible Programming Environments

The accessibility of the activity affects children’s efficacy in learning and during collaborative interactions between them [21]. Thus, when designing a coding kit, it should be taken special attention to the accessibility of materials. In fact, when using robots or any other technologies to teach CT to children, they tend to be very pleasing visually, through the well-made use of aesthetics and the constant feedback given by the interpreter of the activity (who can either be a robot or any other character in an application). However, for children with visual impairments the adaptation of the usability - to program actions - usually tend to damage the environment, making the story or the composition of the activity poorer [22].
Therefore some experiments considered the accessibility of the activities and tested their efficiency. In fact, blocks have been used in substituting (input) software (which avoids complicated errors like the ones that are related to syntax), and robots as the output hardware, which respond to children's orders [10]. These block-based environments use metaphors where operations, variables and constants are conceptualized as “blocks” - puzzle-pieces that can be dragged and dropped into a program. Virtual programs usually took into consideration syntactically correct codes to remove syntax complexities (so they only allow the assembly of blocks when it is correct) [16].

Plus, when designing environments for children with visual impairments, that work with virtual programs and robots, some accessibility aspects should be considered. For example, some edges and corners of visual elements need to be reinforced; the precision for selecting objects should be reduced (by increasing the size of buttons, for example); touching the screen one time does not accidentally change it; after being selected, elements/actions should be described; spatial layouts should be reproduced when possible; the number of gestures requested by the app should be reduced (since children have difficulty reproducing them in a way that the screen reader understood). Thus, Blocks4All [16] explored the 5 barriers for already existent block-based virtual environments: the perception of the output (purely visual or also audible and tangible), of blocks and menus (it was impossible to focus on blocks in the toolbox or in the workspace using a screen reader), of the movement of blocks (in most apps, by using drag and drop), of the program structure (is it understandable when the blocks were nested (inside ifs or loops)), and of the type of information (data types conveyed by blocks colours and shapes). To correct these mistakes, Blocky used a robot (DASH) for the output, making it accessible to children. Besides, it made elements accessible using voice commands to give cues on how to manipulate blocks. Plus, it added a Zoom feature to check the toolbox and the workspace. Also, the user had two alternatives for moving the blocks: through drag and drop voice-guided or by selecting a place and a block. To check its structure, the program was visually or orally transmitted in a hierarchical list. Finally, it supported three types of blocks: operations, numbers and Booleans (and they played different sounds making the blocks on the workspace respond to them with similar audios). This experience has shown that all children were capable of listening to the robot and following its movements. However, VI children only easily accessed blocks when they were resizable, aligned in the screen’s bottom and separated by white stroke. Plus, children with low vision did not use the Zoom function, but narration instead. Plus, none of the children used “drag and drops” correctly, and they preferred to choose the block first, instead of "location". Thus, this experiment has shown the importance of conveying information in multiple ways [16].

However, by playing with physical objects with symbolic meaning manipulation, children explored more complex symbolic thinking [10]. The experience [24] demonstrated the role of designing aspects of embodiment (embodied - activities that involve body action/physicality) in the facilitation of intuitive interaction in children in the context of tactile interactions. When designing for children, it is essential to
make easy for them to learn with behaviours that they can (sub)conscious anticipate, giving priority to previous sensorimotor knowledge derived from embodied interactions with the physical world, acquired from the environment through the sensory, motor and kinesthetic systems. The epistemic (ways in which people learn about their environments, helping them deciding their future actions) interactions may not offer any information about a solution to the problem, but they help reach the goal by revealing clues in the environment (for instance, completing a puzzle). Therefore, physical Jenga requested physical affordance, which is much more intuitive than perceived affordance used to interact with virtual interfaces. In fact, natural cues on physical Jenga like spatial orientation, material properties, e.g. were using children's past experiences. On the other hand, contextual cues on virtual Jenga were not perceived because they did not tap relevant sensory-motor or experiential knowledge. This positively reinforces the work with physical blocks, in detriment with the virtual ones, even for sighted children.

In fact, during the experience with DASH and its robot-based programming environment [26], tangibles took an important role, since they were recognized in reducing cognitive demand (because thinking is linked with motor actions and manipulating blocks activates processes that "may serve to integrate conceptual knowledge"). These blocks were literally conceptualizing the robot’s moving directions, so they facilitated the understanding of abstract concepts. However, not all of them were well perceived. While the 3D arrow symbol blocks used to program the robot were quickly understood (as children made easily the association with the arrow and the direction the robot would take), action blocks were less intuitive. Nonetheless, the Play block had an increased value, as it allowed children to use other blocks anywhere on the table. This has created a "working area" that has shown high relevance for visually impaired children, because they could manipulate objects as needed, to understand better what they should do (on how should they manipulate the robot) to perform the tasks.

Another element that suggests improvements while teaching programming to children is the use of maps, since they also increased independence, from other aspects. The following experience [26] addressed the topic of accessible maps and clarified designing keys for making them. First, it explained the difference between 2D Tactile and 3D maps. While the former is referred to raised line drawings and required some experience to be used and understood (although it usually has simplified content), the latter required less effort to be read and enhanced understanding with its 3D printed models, helping identify and memorise routes. Plus, it gave a positive message about inclusion and accessibility, capturing attention from VI and sighted communities. In fact, 3D icons were enjoyable and engaging. Moreover, 3D maps that included audio descriptions increased spatial memorization. On the other hand, the number of distinct symbols/figures should be reduced between 10 to 15, to be discerned from one another. Plus, the design must be simple enough to be recognised, while considering that the objects would probably not be completely analysed. In fact, most people felt the top of the figure in more detail and neglect the bottom. However, even when roads and pathways were easily understood, it was suggested that
they were wider than 7 mm, to be followed with full finger pad. Plus, the use of meaningful colours also assisted with decoding. Although it was suggested that the elements of the map were labelled by braille or audio key, since braille is not readable by every VI people, and has a less evident benefit [21] - even because few blind people are literate in braille -, audio labels were defined as better solutions [26].

The DASH experience used a map composed by EVA foam tiles, with two interleaved different colours to enable children with low vision to distinguish them [6, 15]. Besides, tiles union was also perceived by touch, so children could count how many were they. To improve the experience, educators also suggested using tactile cues such as reliefs for example, as they believed children with visual impairments would better perceive them. In fact, in another study, the differentiation between squares was suggested to be made by raised lines rather than engraved ones [15]. Another experience suggested making the contrast using different textures, since children benefited from an "organizing system" that used different types of plastics for each type of blocks, providing a logical and tangible structure for finding blocks and parameters [21].

Plus, DASH also integrated feed-forward sounds which made children anticipating its actions [24, 26]. This resulted in them feeling safe, and under control of the situation, [8], which increased their interest in the robot and made them easily understanding concepts associated with the experience. However, the audio cues had to be carefully designed, to avoid cognitive overload and jeopardized learning [6], but they should be included since children from these studies tend to like to have audio integrated with the activities [27], and they were useful for them, to inform about the current scenario when they are asked to [14]. Moreover, educators suggested to also integrate vocalized interventions to the robot (a failure would make it vocalize a sad interjection; while reaching its goal would make him proclaim victory), or even contextualized sounds (like birds, ocean waves, etc.) [8].

Finally, given human aspects - either physically and in personality - to the robot turned it more accessible to children. In fact, previous works have approached "emotional" robots and children reactions to it. With a pet robot that expressed seemingly autonomous emotions, the user expressed a feeling of familiarity. However, a robot which sympathizes with children had sharpened positive impacts in terms of the long-term interaction between them, and emotion-expression methods are beneficial for human-agent interactions (at least better than models that express random emotions). Therefore, this study [9] has shown that robots should be designed to have something in common with their users; otherwise, children gradually lose interest in the robots as learning progresses. Thus, to foster collaboration between users and robots, programming the robot to sympathize with the learner and express the same emotions as they do, instead of using a robot that uses an emotion-expression method alone without considering the learner, was more effective. This study proposed a sympathy expression method for robots to express sympathetic emotions during interactions with learners. To do so, the robot reacted to correct and incorrect answers taken by the children, using a pre-defined spectrum with emotional faces/reactions.
By increasing the number of right/wrong answers, joy/sad reactions also increased their levels. Plus, as it gave verbal hints and reacted to children’s answers, it gave them the impression that it was evaluating/monitoring them (like humans instructors do). However, when the robot was incongruent with its empathetic behaviour, it caused negative effects on the user’s trust. Finally, the collaborative learning scenario with a robot which alternately solves questions with a human while expressing emotions, although it promoted more learning than the collaboration between children, it had a larger period of time of they wanted to play, than when the robot expressed random emotions [9].

In another study, to understand what children expected from the robot, they were asked to define its features: physical appearance, facial expression, personality and character. Some of the suggestions were making the robot a child professor/assistant, that taught and played with them. Plus, it expressed positive emotions through its voice, cartoon face and luminous parts. While most children did not draw noses on their sketches, mouths were always present, to emphasize emotions. They imagined this robot to be either male or neutral gender, and most groups have drawn clothes and accessories to it, and passive and replaceable arms [28].

Plus, reinforcing the importance of humanizing the robot, the experience with Dash has shown that by giving human characteristics to it (it was given eyebrows to DASH, as well as the "capacity" of speaking and dancing), children become more predisposed to establish an effective relationship with it. As a matter of fact, after the experience, educators highlighted that children may have created an emotional bonding with the robot and because they were learning to program it with pleasure, they may also be delighted for future learning activities [26].

Having these concepts on how to improve accessibility for VI children, the following experiences also addressed this question: Even though KIBO’s experience did not consider VI children, the kit itself can be rearranged to make it accessible to them, needing only some adjustments. In fact, both input (blocks) and output (robot) were tangible pieces, which usually makes the participation in the activities easier for VI children. But, since the blocks did not have tangible cues that differentiated them from each other, they needed to be upgraded to make it a more inclusive activity. Plus, as the experience showed that even visual children had some difficulties using KIBO to scan their sequences, another improvement to counter this would also be necessary, to avoid frustration. Plus, it integrated some audio feedback (like the wheels moving) which gave some spatial cues about the robot position [10].

Just like Kibo, Bee-bot experience was not designed to integrate VI children [2], however, it also had the tangible components for input (the buttons on its back, which have 2d raised lines that indicate the forms of its drawings) and for the output (the robot itself). As it had sound while moving, it also gave the perception of where it was. However, for the map, the author does not explicitly says if the labyrinth had tangible components or not (but the image it presents suggests that it did not have it). So, to make it more accessible, the labyrinth could be 3D impressed.
The Cellulo experience was accessible to both children, with and without Visual Impairments. In fact, the robot had haptic feedback that collaborated with children by showing them where they should have been directing the robot to make the requested shapes of the activities. Although it played different sounds for different phases of the activities (start point, change of the stroke), none understood. Therefore, the audio feedback should be readjusted accordingly to children's expectations. Besides, it also had visual feedback - the different LED's' colours on the robot showed different states - but the results showed that it was not efficient for children with visual impairments. Plus, visual, haptic and audio cues had better performances when used together [5].

The Hungry Cat is an audio/haptic feedback based game for children with visual impairments, easy to set up and use. Objects/Obstacles were identified when an interaction occurred through audio and haptic feedback provided by the environment. However, it had some problems, not making it completely accessible. For example, to move the Cat, the user passed on his actions with the tablet - moved it forward by tinting the tablet forward; moved it backwards by tilting it backwards; turned the cat by turning himself while holding the device. This game also included audio cues such as telling the user about its state (like "You are facing a wall" indication, or indicating how far the user is from his objective); sound effects and pre-generated speech. The sound was louder if the user was near to the objective. Plus, it was place oriented - the sound was reproduced on the left/right ear if it came from left/right in the game. Haptic cues like vibration were enabled when the user bumps into something or for simply counting steps, for instance [18].

Torino was modelled for having special attention to audio feedback, and physical representations (identifiable through touch) [4]. To create computer programs, children assembled instructions "beads". There were 3 types of them: play, pause and loop, distinguishable by size, shape, and by their physical controllers that use contrasting colours. Also, the work-space used black felt mat and had a storage box with different types of compartments, specifically for each bead type, which facilitated spatial orientation and the localization of spare beads. Children identified bead types and understood their functionality not only through their physical distinctions but with a combination of tactile clicks and its response in audio feedback. By handling and sharing components, VI children were capable of identifying program instructions, manipulating controls and assembling programs. However, Torino had a problem with audio feedback. Because every sound came out from Hub, it may have caused some confusion to children when, for example, instead of unplugging 1 bead, a sequence of them, or 2 beads (from different branches) were unplugged at the same time. Plus, the system did not differentiate if a bead was wittingly unplugged or if the link was accidentally loose. Thus, it should have provided localized information while keeping the complexity of sounds significant and manageable.

When designing inclusive games, they should be accessible for both children with and without visual impairments. This means that the game should not be simplified (non-engaging) to the point of forget-
ting that sighted children are also participating in the activities. Thus, the kit should consider tangible cues and contrasts (colour), but it should also be colourful and have visual images/symbols. During the first workshop activities, children reflected Ozobots’ accessibility for children with visual impairments and they were asked to propose solutions to the identified issues. They found 3 main problems: it was difficult to locate on-off button (so it was augmented with rubber, to be more easily differentiated); it was complicated to differentiate between robots, so children customized them using crafts, and also, accessing line drawings and following the robots as they were moving (so they were augmented using tactile tape). Therefore, given the mixed visual abilities children present during the experience, multi-sensory materials and feedback were essential to engage all participants. Besides, using crafting materials to customize robots and design maps were also engaging and narration to guide and design activities was important to moderate pace [1].

2.3 Collaborative Learning

Collaborative learning is a learning situation where participants share efforts to reach a mutual learning goal. For complex tasks, communication - “trying to get a common and shared representation of the situation” -, and coordination - “to make a plan on how to solve the case” - are crucial for the success of collaboration [17]. In fact, communication and collaboration supported the teaching of CT, as they are 2 of the main aspects to take into consideration when adopting PTD (Positive Technological Development) framework, that describes some positive behaviours used while teaching robotics programming [2]. Plus, when collaborating while learning, children can reach higher achievements and productivity [29], foster caring, supportive and committed relationships, aid psychological health, self-esteem, and social competence.

However, the lack of vision sets a barrier in non-verbal behaviours cues intrinsic to social communication and coordination of tasks [4]. The age of the children also have an impact since during some experiences with robots, by analysing different aged children while they were solving tasks, it was made clear that even tho they could work with the programming activities, second graders were more likely to work collaboratively than first graders [11]. Plus, the object and engagement to the activity also have important roles to foster collaboration [4] and tangibles revealed their importance since while teaching CT with them, children got into a playful learning mode and collaborated (by dividing, exploring and sharing) [6]. Thus, the accessibility of the materials influences collaboration between parties [21].

Thus, besides being an important element for developing CT (as seen before), story mapping is a collaborative activity that supports creativity and facilitates group interaction, used to promote children’s understanding of stories and narrative structure. Also, it promotes imagination, builds confidence and improves memory and sequencing skills. This paper [15] shows that using multi-sensory materials fosters
interaction, participation and group discussion. Also, it facilitated a collaborative exploration/explanation amongst children with visual impairments.

Moreover, as tangible outputs, robots can run interactions with children and promote social interaction between them [30]. As working with robots while programming is fun, it can be used not only to increase motivation but also to improve collaboration [2]. In fact, using a tangible robot-based programming environment for space activities promoted inclusiveness and educators saw the robot as a motivating artefact that could reinforce casual relations [6]. Plus, in the study with Bee-Bot, educators agreed that activities that used robotics develop collaborative learning [2].

In this context, the Dash experience showed that even tho collaboration was not fostered; it was present. In fact, children tended to support each other, by instructing and reinforcing - they were guiding each other to where the robot was and correcting the blocks or the arrow’s direction, etc. However, collaboration was broken when a peer took to long to perform an activity. To foster collaboration, educators suggested dividing different blocks to each child/group, which would force it to solve the activity [6]. In fact, by attributing different roles to each child, collaboration can be fostered [15]. Therefore, educators could have used strategies like this of assigning roles to each person, giving them responsibilities; or setting tasks that can only be achieved when working together; encourage the person to seek and give help; explain their actions; reflect about alternative approaches to solve problems with their partners; etc.

During the experience with Torino, they used a process of jointly (between children with and without visual impairments) exploring and discovering physical and auditory features of each component, which has resulted in them sharing the same “vocabulary” and points of reference, crucial to foster collaboration between children with and without visual impairments [4, 17]. In fact, all groups bound up their interactions through conversation. Children with (some) vision visually monitored each other’s actions by pointing or laid out particular beads as references when explaining a problem or plan. Blind children used audio feedback, the manipulation of objects and making contact/guiding their partner’s hand to express themselves [4]. During the experience, one of the children said to another that his bead manipulations were inaudible during parallel interactions. This has resulted in developing a more productive turn-taking approach: by explaining one’s idea or actions, and asking the partner for advice it served to introduce alternative problem-solving approaches and highlighted individual strengths; building on which advanced their program and process of learning. Thus, this activity fomented play, creativity and a sense of progress, which contributed to positive learning experiences.

Another suggested alternative was to make groups focus on competition - considering the mixed vision abilities of the group, to make possible anyone achieve the solution -, to see “who would get first into a solution?”. However, it was established that it would have been detrimental for collaboration to use small boards/trays unless there were different boards for each child while working together to reach
a common solution. Also, having a map on the floor could have compromised sharing the programmer role between children. Thus, some educators defended putting a small map on the tables [6].

As previous work has shown when collaborating to solve tasks/activities, children with and without visual impairments faced the problem of not sharing the same sensory perception of the task. So, the use of multi-sensory interactions have positive impacts on both parties reaching a middle ground of understanding. Similarly with “Robot Hokey Pokey” [12], during this experience with Cellulo, there was negotiation, collaboration, dialogues and shared responsibilities and the qualitative analysis revealed positive engagement and collaboration between students with and without visual impairments [5]. The experience with Kibo also reinforces this idea. It showed that children were more engaged to collaborate (group discussions within small groups or the full class) by introducing new concepts with songs, dances, games or storytelling. In fact, educators said that both communication and collaboration were 2 of the main social aspects highly promoted during the experience, followed by content creation and creativity [10].

Also, the Ozobot experience reinforces that, when designed with inclusion in mind, games should promote learning and interactions that are engaging to both sighted and VI children. Thus, by making the coding kit appealing for both, children would collaborate more easily [1]. In fact, they enjoyed collaborating in pairs to code the robots, using coloured pens and colour-coded labels, and the games’ competitive nature. Plus, they really liked building and traversing the obstacles course - creating maps and sharing instructions were a means for focused and engaged coordination and learning. Moreover, by doing activities with others and giving all children the opportunity to take/swap roles, it widened perspectives, promoted empathy and fostered inclusion and the consideration about each other’s needs, abilities, perspectives, etc. [1].

However, if to teach CT robots have been turned into moderators of the group, 2 functions can be used: performance-equalizing (increases task performance but decreases group cohesion), and performance-reinforcing (the opposite). Because social and task goals of a group can be opposed to each other, the study has shown that long-term social benefits of a moderator robot, offset short-term decreased in task performance. Plus, a moderator robot can positively affect the social dynamics of a group [25]. To communicate with children, Baxter used gestures. However, this experience took into account deaf-blind children, and so their communication channel remained haptic. However, human-robot communication can be enhanced using sounds, utterances, and lights [30].

On the other hand, the study [17] relied on a robot being a mediation tool for an activity based on a collaborative treasure hunt game, between two people with visual impairments. Its objective was to understand if Cellulo would ease the task and decrease the time needed to find the treasure while helping the collaborators reach collaborative optimal effort and better co-construct knowledge about the space. The game was composed by two children, an explorer (on the field) who was following
instructions and a guide (who was remotely giving them) that was on another room; and it was tested by 2 different perspectives: active - where Cellulo moved (impersonated by the technical team) on the map according to the guide child instructions; passive - where it was only a printed block, moved by the guide child. The results showed that active Cellulo helped children achieve better performances than when the robot was in passive mode. In the latter, guides struggled with it, as it was representing “another element to deal with” and used their own hands to represent the explorer. Therefore, information processing - “importance for participants to process and share complementary knowledge about the spatial layout of the game” - , interpersonal relationship - “evaluates the behaviour of both partners during the game” - and motivation - “illustrates the commitment of each person in solving the problem” - were impacted with Active VS Passive modes. While the passive condition was better at supporting optimal collaborative effort (since both children had to be much more coordinated), the active improved the task’s performance.

### 2.4 Discussion

Section 2.1 highlights the importance of teaching CT in early ages, because of the benefits it gives to children while they are growing (as STEM promotion, for instance [31]). It is even more crucial for visually impaired children since it develops other skills such as space notion and navigation, making them feel more autonomous and confident. However, since collaboration also brings positive effects on children’s lives - if on the one hand, it counters the feeling of isolation; on the other hand, it fosters higher levels of productivity and improves their social competences. Nevertheless, if the environment is not accessible to everyone, it can have the opposite effect and bring a feeling of isolation to VI children if they feel they cannot work together to achieve a task’s goal; or frustration to their sighted peers, due to its simplicity.

Thus, Table 2.1 summarises the different approaches addressed in the previous sections (3.1, 3.2
As Computational Thinking is directly linked to programming activities, it can be fostered using “coding robots” activities. Besides, maps upgrade them and challenge children to use navigation and spatial perception (essential for VI children). Plus, for making accessible activities, input and output must rely on tangible elements, to be perceived by both children, with different textures (to be distinguishable and grouped, so VI children easily identify each piece). However, it cannot neglect appealing visual elements as they are so crucial to engage children without visual impairments. Plus, additional features as sound effects and audible feedback also attracts both children. Finally, collaboration can be naturally fostered during accessible and engaging activities; or encouraged by setting tasks with objectives, giving roles to children or even through competition.

However, visually impaired children and their sighted peers addressing the same activity for the first time do not share the same referential, making communication between them more challenging. This issue can be overcome forming working groups and introducing new concepts with songs, dances, crafting, games or storytelling, e.g. Because they have time to explore the robot and how it works together, they end up sharing vocabulary. Another way to address this question is following the concept of the experience with Cellulo and not giving children the perception about the other’s workspace, but just defining an equal reference point before the activities start. This will set them up in a similar balanced situation where the other’s working area is unknown, and they will have to collaborate to reach the activity’s goal.

Moreover, the table shows that almost all the experiences required children’s physical presence, which complicates installing the before solution because of their physical proximity. Thus, the exploration of remote solutions for implementing the activities remains untapped, even tho they could enable this balanced start for children. Moreover, it would let them explore CT and collaboration at leisure or even with educators during remote classes (critical in these pandemic times), opening doors for adopting collaborative and engaging activities anywhere and anytime.
3

Accessible Remote Robotic Kit - Designing Process

Contents

3.1 Game Designing .......................................................... 27
3.2 Sokoban Map .............................................................. 30
3.3 Instruction Blocks ....................................................... 33
Considering the stated advantages in using blocks to teach computational concepts, and robots for fomenting collaboration between mixed visual abilities’ children, we wanted to adopt a design for our Remote Coding Kit that uses Ozobot Evo (a small robot), and small 3D printed blocks.

Besides, in several studies it was pointed out that existing systems could be used to carry out non-school-related activities, while still teaching children CT key concepts. Thus, we decided to design a system to be a playful activity [2,18,22], like a game, where the main character would be played by the robot, with its actions translated with directional blocks.

However, as mentioned in chapter 1, section 1.4, we wanted to balance children’s knowledge about the activity. So, we would incorporate sound and textures into our kit to give children more than just visual information. However, to balance the number of information channels between children, we needed to test attributing player roles to them. Suppose both of them have the opportunity to address both roles in a remote context (which divides their working spaces and reduces the visual information for sighted children). In that case, the system can become more balanced.

### 3.1 Game Designing

To decide on a game, we started looking for the ones that could be adapted from virtual interfaces to become tangible, to be accessible to all children. As mentioned before, they should also have a main character that could be played by the robot, and whose actions could be controlled using directional blocks. Furthermore, since one of the goals of our system is to foster computational thinking, we searched among games that belonged to the puzzle category [32]. We ended up divided between two different games for our system:

**Maze:** The objective of this game is to **move the character from an entrance to a goal**. So, the challenge is to analyze the map in detail, to choose a path that does not have its passage blocked by a wall. Usually, the larger the map, the more difficult it will be to choose the right path without mistakes, since larger maps means more path possibilities. In this game, a block (with a directional arrow) can directly affect the objective, since it would translate the character’s movement.

**Sokoban:** The objective of this game is to **use a character to push a crate to a storage location**. Its complexity relies (among others) on the branching factor (e.g., the number of “children” at each node). It not only means that for the same end, it can have different paths, but also that the more subsequent actions the map requires, the more difficult it will be to solve [33]. Therefore, although the character movement is what is being controlled, it is secondary. Thus, children have to position themselves on the crate’s perspective and understand the path it has to take, to finally use the character to push it into the final position. In this game, a block (with an arrow) can not directly affect the objective, since it would not translate the crate’s movement.
With the Maze, the simple control of the robot from an entrance to a goal could be not engaging enough for children, while controlling it with the objective of the sokoban game could be too much complex for their ages. Thus, before developing our system, we needed to validate:

- If the games we proposed were engaging for children with ages between ten and thirteen years old;
- If the division of the system’s knowledge between them (through the attribution of roles to each child and the imposition of the remote factor), was balanced while still captivating for them.

### 3.1.1 Formative User Study

We conducted a formative user study with three pairs of sighted children, where each pair was challenged to remotely collaborate (using Zoom) to reach the solutions for the two online games we have chosen: Maze ([https://www.mathsisfun.com/games/mazes.html](https://www.mathsisfun.com/games/mazes.html)) and Sokoban ([https://www.cbc.ca/kids/games/play/sokoban](https://www.cbc.ca/kids/games/play/sokoban)).

For each game we have selected three levels with increasing difficulties. The first level of both games was played individually so that each child understood the goals and functioning of the games and got familiarized with the system [1, 10]. On the second level, they started to play together. However, their knowledge on the system was separated, as we assigned roles to them: they could either be the Map Explorer (responsible for understanding the map and seek for a solution) or the Block Commander (that had to use arrows and showed them to the investigator, to make the character move). Therefore, there was no longer a direct control over each other’s workspace and they could only influence each other through dialogue (using Zoom). Furthermore, both children needed each other to solve the levels, as without the instruction’s arrows the character would not move, and without the map, it had no concrete objective to accomplish. Finally, for the third level children switched roles.

After completing all levels, children answered a simple questionnaire about how engaging and difficult the activity was.

### 3.1.2 Findings and Discussion

When asked if they preferred to play both games with another player or alone, all children agreed they preferred to play with another child. Some of them even reinforced that on the question of what they have liked more.

None of the children solved the Sokoban second and third levels at their first attempted (without committing mistakes). In fact, most of them got it wrong more than once. But the second level of Sokoban had a shorter average resolution time, than the second level of the Maze.
Although all children agreed that the Sokoban was more difficult to solve than the Maze, when asked about their preference, they were divided - three answered that they preferred the Maze game, while the other three preferred the Sokoban game. In both cases, the reason was the increased difficulty that Sokoban presented.

However, children’s ratings about the increasing difficulty were not considerably high (from 1 to 5, five children voted below or equal to 3 and only one said it was a 4). Besides, the first pair of children (who were close friends and thirteen years old) quieted the third level of the Maze game, expressing some dislike for it being too long.

Finally, none of the children showed annoyance at being in charge of the blocks. In fact, when asked what they liked best, one of them responded that they preferred to have this role during the activity, and none of them said they preferred to be in charge of the map.

### 3.1.3 Conclusions

In general, the children were communicative. The assignment of roles was not inconvenient, and the children enjoyed playing with each other.

The study showed that the games were equivalent in terms of children’s involvement. However, as
mentioned above, the first group dropped out of the third level of the Maze, because it was too long. However, its size was what would be able to increase its complexity (since our blocks had only the four possible directions that the character could take). Therefore, as children showed annoyance about this factor, we felt that we should not use it to develop the kit. On the other hand, the complexity of Sokoban was what made them split in terms of their games’ preference.

So, we decided to use Sokoban, but lower the complexity of the levels. So, the challenging side of the game that some children said they liked remained - that is, the allocentric perspective that Sokoban provides, rather than the egocentric one from Maze - and, at the same time, we were circumventing the problem of them feeling that the game would be too difficult to be solved. Furthermore, since children with visual problems tend to use more egocentric than allocentric representations [17], Sokoban could be a good challenge to help them develop spatial awareness and computational thinking.

3.2 Sokoban Map

Intending to use the Sokoban game, we needed to transform it, from the virtual application we used, to become a tangible game, to be accessible to mixed-visual ability children.

3.2.1 Ozobot Pushers

We have tried the Ozobot pushing a crate with three different pusher models, and we have chosen the third pusher (https://www.thingiverse.com/thing:2218396) since its crate contact base was more significant than the second one (https://www.thingiverse.com/thing:2975591), and it did not drag the crate each time the robot turned, as the first one did (https://www.thingiverse.com/thing:3542215). It was printed in white to camouflage with the robot and match its starting position on the map.

![Figure 3.2: The image shows the three different 3D Printed Ozobot Pushers in chronological order.](image-url)
3.2.2 Crates

We have chosen the design of the boxes from https://www.thingiverse.com/thing:2975591, as shown in Figure 3.3. However, we resized them to have x cm and x g and choose an infill density of 15

and decided to 3D printed them in yellow, to make contrast with the green of the floor, while match with its ending position.

![Figure 3.3: The image shows the 3D printed Crates.](image)

3.2.3 Map: Walls and Floor

For the map, we used plates and blocks from LEGO, to make the walls and obstacles. We took the idea from another study [34]. Not only because they are resistant, but also because children frequently use them so that they would be familiarized with them. In the referred study, the robot walked over 3D printed caps made explicitly for LEGO plates. Although we had access to the design of these caps, we had to reformat them, as the robot that had been previously used was smaller, and ours could not fit between LEGO walls, with the distance of one "2x2" cap. Consequently, we tried to make 3x3 caps, but the robot pusher still bumped into the walls when trying to change direction. Finally, we ended with a 4x4 version of these caps.

After we arrived at a general design that covered the plate floor and allowed the robot to move freely, it was necessary to create specific designs that would give tactile and visual clues to differentiate the different floors according to their functions. We left the “path” cap green, as it was on the other study. However, we still needed three more designs to line off start and end positions of the robot and the crate. Therefore, we used three colours that contrasted with green and made:

- A white cap with a cut circle that marked the robot's initial position (and direction);
- A black cap (more discreet and less relevant during game-play) with a square that marked the starting position of the box;
- A yellow cap (vivid colour) with an "X" that marked the final position of the box.
The grooves did not give enough tactile cues for the caps to be distinguished from one another. Therefore, we put some material that contrasted with the feel of the piece’s material. For the white cap, we used self-adhesive paper and cut through the grooves in the piece. For the black cap, we started by trying to put electrician tape (because it is black and soft) on the square cap. However, this tape caused more friction and atrophied the robot’s movements despite feeling better to the touch. So we switched to the same type of paper sticker cover that we used on the white cap. Because the “X” from the yellow cap had more grooves than the other two, it was the most distinguishable one. So it remained without any other cue. Finally, to make the pieces more appealing and highlight these cues, we have outlined the grooves of the yellow and white piece with a black marker.

3.2.4 Map Levels

Children completed the first level of the game (Figure 3.6) without any significant problems. Therefore, we decided to keep it for our activity.

To lower the difficulty of the other levels, we started calculating the number of instructions and iterations each level had, as shown in Figure 3.7.

Then, to not disperse children’s focus to various problems simultaneously, we limited our design to
one Crate per level. The complexity of the levels would increase with the increasing number of iterations and instructions, so we designed the new second level with just one iteration but with more instructions than the first one. The third level had two iterations and even a more significant number of instructions than the second one. Nevertheless, we made the first iteration with only one possible solution to facilitate level resolution. The Figure 3.8 represents our sketch for the levels.

Unlike the virtual application, in which a "Left" instruction makes the character move to the left, our "Left" block would only make the character turn to the left.

### 3.3 Instruction Blocks

Although there have been another attempts to make instruction blocks with LEGO and TopCodes [20, 34], we discarded that option because they were too large and the symbols from TopCodes only had meaning for the program and not the children. However, we wanted to keep the shape of the k-line blocks from the study [20], but 3D print them, to make them more resistant.

So, in order to make four types of blocks (to move "Back" and "Forward" and to turn "Left" and "Right"), we needed to design four different blocks and choose four different colours. Besides, we also needed a plate for the children to assembly the instructions they wanted to apply.

Figure 3.8 shows our attempts to design the plate. The first trial did not work because if the blocks were crooked, the sequence was poorly read. So, we forced the sequence to be read by the block’s height position. Then, we improved its efficiency by making two slots instead of one. However, the blocks could be placed upside down due to the symmetrical design of the pins.

So, we ended with a plate with a three-pin per block design and blocks with directional arrows and 3 holes to fit into the plate Figure 3.9(b).
Figure 3.7: The image shows respectively the second and third levels of Sokoban, from the first study we made. The second level needs at least two iterations, with a total of thirteen instructions given to the character. The third level needs at least six iterations, with a total of thirty four instructions given to the character.

3.3.1 Colour Preference Questionnaire

To choose the blocks colouration, we made a questionnaire where we analyzed the general preference of contrast and luminosity on four different coloured blocks.

There was a total of forty-eight people answering our questionnaire. Twenty-four were between twenty-four and twenty-nine years old. Seventeen were between nineteen and twenty-three years old, and five were between fifteen and eighteen years old. Finally, two had more than thirty years old.

Twenty people responded that they have a type of vision difficulty and the most answered problem were myopia and astigmatism.

When asked individually, most people preferred light green, medium blue, dark red and medium yellow. However, most people preferred the first set of blocks with medium lighted colours and when asked to form a new set of colours, the most voted ones were also the medium lighted ones. Therefore, we used the most voted colours for the 3D printed blocks.
Figure 3.8: The image shows our design for the second and third levels of Sokoban.

(a) Plate Chronological Designs.  
(b) Blocks’ Final Designs.

Figure 3.9: Plate and Blocks’ Designs
Architecture

Contents

4.1 Web Application ......................................................... 39
4.2 Block Commander ...................................................... 42
4.3 Map Explorer ............................................................ 44
4.4 Keyboard Cover and Screen Reader ................................. 45
The attribution of roles in the first study succeeded in balancing children's knowledge about the system, and it was engaging for them. However, during the study, we played the role of the computer by analyzing the arrows and making the character replicate the actions. So, we needed to automate the transition of this information from one player to another. For that purpose, we decided to make a simple web application that is further explained in this chapter.

Figure 4.1: General Architecture of our remote robotic kit. Block Commander on the left and Map explorer on the right

The Figure 4.1 represents a general architecture of our remote robotic kit that divides the system into three "entities": two clients and a server. The Block Commander client assembles the instruction blocks on the board inside the “Magic Box”. A camera located on top of the Magic Box reads these blocks to the laptop. After their registration and acknowledgement (given by the Block Commander), the web application sends the photo to the server to be analyzed and translated into a command string, later sent to the Map Explorer client. Finally, the Map Explorer will accept send these commands to the robot via Bluetooth.

4.1 Web Application

As mentioned above, the web application ensures communication between the Block Commander and the Map Explorer. We made it with a dark background to contrast with the colours of its buttons and the images. The Figure 4.2 shows the first menu with two big buttons corresponding to each player's roles, selectable using the “TAB” key, and when the wanted button gets highlighted, “Enter”.
4.1.1 Block Commander

The Block Commander page has a static menu that explains the available keys’ functions and a sight of the inside of the Magic Box. With “Spacebar”, the user registers the instructions of the Blocks: the image is sent to the server and converted into a command string that returns to the client. The commands appear written on the web page for debugging purposes since it allows the user to confirm the instructions on the plate before sending them to the other player (by clicking on “Enter”).

Figure 4.3 shows a diagram with the HTTP messages exchanged between the server and the Block Commander Client.

4.1.2 Map Explorer

The Map Explorer page starts with an animated SVG image that holds while waiting for the instructions. Once the server sends them to this client, a new static menu replaces the SVG. If “Enter” is clicked, the commands (sent by the Blocks Commander) are saved into the computer. The Ozobot will further execute them, and the menu switches back to the SVG. Figure 4.4 shows a diagram with the HTTP messages exchanged between the server and the Map Explorer Client.

4.1.3 Presential User

The web application also allows two users to play presently together (accessible by clicking on “Esc” on the the first menu of the App). The web page is the same as the Block Commander one. However, “Enter” saves the commands on the clients’ laptop (instead of sending it to another one).
4.1.4 Implementation Details

The web application was developed using a Python based framework - Django -, and its templates were created using HTML, CSS and JavaScript. It was deployed on an Heroku Server (https://thesis-collab.herokuapp.com) and the communication client-server and server-client was made through AJAX POST and GET requests and responses.

When the camera on the "Magic Box" captures an image, it gives the user audio feedback, and the image is transformed into a data URI. It contains a representation of the image submitted to the server via an AJAX POST request. The server decodes the image and saves it on an internal directory. Then, using the OpenCV library, we made the server analyze the image. It finds the blocks’ contours and orders them based on their xx and yy centres’ position (from top to bottom, and from left to right). Then, it crops each block from the original image to save them separately on the same directory, to be analyzed to discover its dominant colour (which colour has the most significant number of pixels). Thus, if it is Yellow, then it is a "Front" block but, if it is Red, it is a "Back" block. If Blue is the most present colour, it is a "Right" block, while if it is Green, it is a "Left" block. Finally, the commands are saved into a string, sent back to the client and shown to the user.

If the Block Commander presses "Enter", the commands are sent to the server, and saved on an internal text file.

To download that text file, the Map Explorer presses "Enter". Then, the script that is running on
this client's computer detects the appearance of a file named "download.txt" in the "Downloads" folder. Finally, it analyzes the file and sends its commands to the robot, as it is further detailed in this chapter, section 4.3.

This web application also has an option for co-located levels (such as L0, for example). In this case, the exchange of messages is simplified, as the client starts asking for answers for himself and not to be sent to third parties. Therefore, the architectural process of reading the blocks by the "Magic Box" and its commands being sent back and shown to the user remains the same. However, when he presses "Enter", they are saved in a text file on his computer, to be sent directly to the robot (section 4.3).

4.2 Block Commander

4.2.1 Magic Box

Besides the web application, we had to make the already mentioned "Magic Box", which ensures the proper reading of the Blocks. It was created with six black A3 K-line pieces since the light of the led is not reflected inside the box with its dull black colour. Figure 4.5 represents the interior UV Mapping of
Our first attempt to power the LED was with two 1.5V batteries. However, its light was fading as the batteries were discharging. So, we made a different electrical circuit, where we powered the LED with the laptop 5V USB port. Because the LED we have chosen (https://www.ptrobotics.com/varios/6897-branco-led-backlight-module-small-12mm-x-40mm.html) - due to our need and the best quality-price ratio -, only supported a tension of 3V, we reduced the port output to 2.9V with a DC-DC Step-Down Module Converter (https://www.ptrobotics.com/alimentacao/2647-modulo-conversor-dc-dc-step-down-lm2596.html). Figure 4.6 shows a representative diagram of our circuit.

We 3D printed a box to hide and protect the module converter in black (https://cults3d.com/en/3d-model/tool/1m2596-and-1m2596s-buck-converter-tool-less-snap-fit-enclosure-ender-cr-10-2020). It was glued on the top of the "Magic Box" to gather easy access to the investigators, to calibrate the tension if needed.

Figure 4.7 shows our final design of the "Magic Box".
4.2.2 Plate and Blocks

We used a transparent plastic box with a fixed k-line black divider to organise the blocks per colour. The plate where children would assemble them was painted with acrylic dull black to make it invisible for the camera. Besides, we also painted the arrows of the blocks with the same painting to make them more engaging and create contrast, making the arrows distinguishable from the remaining blocks.

4.3 Map Explorer

4.3.1 Robot Movement and Lights

We followed the tutorial from https://codeburst.io/remote-control-ozobot-evo-raspberry-pi-python-bluetooth and attempted to use JavaScript to communicate with the robot. However, despite our success con-
necting with it through Bluetooth, we could not send instructions through its characteristics UUIDs (8903136c-5f13-4548-a885-c58779136702 for driving and 8903136c-5f13-4548-a885-c58779136703 for changing its lights’ colours with JavaScript).

Thus, given the impossibility of sending instructions to Ozobot using the web application, we have decided to make a python script that used an online library (available at https://gitlab.inria.fr/line/aide-group/ozobot-api) that was capable of establishing a Bluetooth connection with the Ozobot and giving it instructions.

The script analyzed the downloaded command string and used the library functions to make the robot move and change its lights accordingly to its instructions. The colour of the lights was synchronized with the colour of the blocks [35] so, for:

- The “Front” command, the robot would move forward for one second, at a velocity of 1200 (units are not explicit on the library), and with static yellow lights turned on.
- The “Back” command the robot would move backwards for one second, at a velocity of 1200, and with static red lights turned on.
- The “Right” command, the robot would spin to the right for one second at a velocity of 290. Besides, its LEDs would increasingly turn on from left to right, with a blue colour.
- The “Left” command, the robot would spin to the left for one second at a velocity of 290. Besides, its LEDs would increasingly turn on from right to left, with a green colour.

Finally, using simpleAudio library, the script would also make the laptop give audio feedback at the end of each interaction to make the system more engaging for children.

We have installed python, pip, bleak (for Bluetooth connection and command transmission) and simpleAudio on the Map Explorer client laptop for running the script.

4.3.2 Maps

The final designs of the maps are shown in fig. 4.9.

4.4 Keyboard Cover and Screen Reader

Finally, we made a keyboard white k-line cover for the laptops that hides the unnecessary keys (Figure 4.10). Besides, we have installed the “Screen Reader” Extension in Google Chrome to guide children during their navigation through our web application.
(a) Level 0 Final Design.  
(b) Level 1 Final Design.  
(c) Level 2 Final Design.

Figure 4.9: Maps Final Designs.

Figure 4.10: Keyboard Cover.
5 User Study

Contents

5.1 Research Question .................................................. 49
5.2 Participants .............................................................. 49
5.3 Robotic Kit ............................................................... 50
5.4 Session Procedure ..................................................... 51
5.5 Data Collection and Analysis ......................................... 52
5.6 Findings ................................................................. 53
5.7 Discussion ............................................................... 57
Previous studies have made engaging, inclusive robotic coding kits. However, even the ones with gaming activities require children’s physical presence, and this condition impedes from collaborating with their friends while solving CT games, anywhere and anytime they want. Besides, since all children have access to the same information, the system’s knowledge is poorly distributed due to the difficulty VI children have in gathering visual information.

Thus, the goal of this study is to understand the differences between the outcomes from our accessible robotic coding kit and a co-located variant.

5.1 Research Question

These studies aimed to address our main research question:

RQ: How does a remote and inclusive programming learning kit with a robot, tangible blocks, and spatial puzzles support Computational Thinking learning and Collaboration compared to a co-located variant?

5.2 Participants

The participants have been recruited from inclusive schools in our country. They were three Special Needs Educator (SNE)s and four pairs of mixed visual abilities children with ages between 10 and 14 years old (Figure 5.1). Two SNEs and four children were from Escola Básica Padre Alberto Neto, Rio de Mouro. The other SNE and the remaining four children were from Escola Básica 2,3 de Cinfães.

Each pair of children was composed by one VI child and one of his sighted best friends.

Table 5.1 summarizes our participants demographic and visual information. Table 5.1 enlightens us to understand their background in terms of programming and previous contacts with robots.

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Child ID</th>
<th>Gender</th>
<th>Age</th>
<th>School Year</th>
<th>Visual Acuity</th>
<th>Assistive Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>C1</td>
<td>Male</td>
<td>13</td>
<td>8th</td>
<td>Blind</td>
<td>NVDA</td>
</tr>
<tr>
<td>G1</td>
<td>C2</td>
<td>Male</td>
<td>13</td>
<td>8th</td>
<td>Sighted</td>
<td>-</td>
</tr>
<tr>
<td>G2</td>
<td>C3</td>
<td>Female</td>
<td>10</td>
<td>5th</td>
<td>Low-Vision</td>
<td>NVDA; Tablet; “Lupa TV”</td>
</tr>
<tr>
<td>G2</td>
<td>C4</td>
<td>Female</td>
<td>10</td>
<td>5th</td>
<td>Sighted</td>
<td>-</td>
</tr>
<tr>
<td>G3</td>
<td>C5</td>
<td>Male</td>
<td>11</td>
<td>6th</td>
<td>Low-Vision</td>
<td>Foot Keyboard</td>
</tr>
<tr>
<td>G3</td>
<td>C6</td>
<td>Male</td>
<td>12</td>
<td>6th</td>
<td>Sighted</td>
<td>-</td>
</tr>
<tr>
<td>G4</td>
<td>C7</td>
<td>Male</td>
<td>12</td>
<td>7th</td>
<td>Low-Vision</td>
<td>Foot Keyboard</td>
</tr>
<tr>
<td>G4</td>
<td>C8</td>
<td>Male</td>
<td>14</td>
<td>7th</td>
<td>Sighted</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.1: Summary of Participants’ Demographic and Visual Information.

Before conducting the studies, we have sent a Consent Form to children’s parents. In this document it was explicit that children would be filmed for further analysis. All parents approved their participation.
<table>
<thead>
<tr>
<th>Group ID</th>
<th>Child ID</th>
<th>Has Programmed Before?</th>
<th>Has Ever had Contact with Robots?</th>
<th>Has Programmed a Robot Before?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>C1</td>
<td>No</td>
<td>Played with a robot that danced</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>Yes: Musical Games</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>G2</td>
<td>C3</td>
<td>No</td>
<td>Yes: Ozobot (from another study)</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>C4</td>
<td>No</td>
<td>Yes: Ozobot (from another study)</td>
<td>No</td>
</tr>
<tr>
<td>G3</td>
<td>C5</td>
<td>No</td>
<td>Yes: Robot Mind with a Map</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C6</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>G4</td>
<td>C7</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>C8</td>
<td></td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>

Table 5.2: Summary of the Participants’ background about programming and robots.

Figure 5.1: Children's Ages - Mean and Standard Deviation.

by signing the form.

5.3 Robotic Kit

We assigned one complete robotic coding kit to each child. We had two per study to simulate the remote scenario without the need of exchanging materials, each time children changed roles. Thus, each robotic coding kit was composed by:

- 1 x "Magic Box’’;
- 1 x Plate to assemble the coding blocks;
- 6 x Yellow Blocks;
- 6 x Red Blocks;
- 5 x Blue Blocks;
- 5 x Green Blocks;
- 2 x Maps - Level 0 and Level 1 or Level 2;
- 1 x Ozobot Evo with a pusher;
• 1 x Crate;
• 1 x Laptop prepared to run the Bluetooth communication script.

Besides, each child’s computer has a computer with connection to the internet, the Bluetooth turned on, and a python script running on the background. Finally, to ensure their communication, we have installed Zoom on the computers and started a meeting on the beginning of each session.

5.4 Session Procedure

We have challenged each pair of children to collaborate to use our system and solve a set of levels of the tangible Sokoban game. We told the SNEs to interrupt the activity if they saw something meaningful that would help children when they were confused or with difficulties.

At the beginning of each session, children were separated into different rooms to solve an experimental level (L0) individually. Similar to previous studies [1, 10], this level aimed to make them familiar with the system: to understand how the game worked and what was the objective (to use the robot to push the crate to the yellow floor).

Therefore, the researchers started by teaching them how to assemble the map (place the Crate above the black squared cap and the Ozobot in the white cut circle, with the pusher aligned with the cut). Then they explained that to move the robot, they would have to assemble the instruction blocks on the plate, make the “Magic Box” read them, and send them to the robot. With the investigator, children made a first attempt for the first iteration with only a “Front” block assembled on the plate. Then, the investigator asked them to complete the level while clarifying their doubts.

After both children had completed the first level, they started the collaborative activity. Thus, we explained to children that they would have roles assigned to them during the rest of the activity, which would define their functions over the system. They could be the “Block Commander” (who would be responsible for assembling the blocks onto the plate and sending them to the other player) or the “Map Explorer” (who had to understand the map and decide a path for the robot to complete their objectives and communicate his ideas with his friend).

As previously mentioned, we designed this study to test the robotic coding kit into two scenarios. In the remote scenario, children were separated into two rooms, and each child had his robotic coding kit. For the presentational scenario, they moved to the same room and used only one kit. Both children experienced being both roles in both scenarios. Besides, we have decided that half of the groups started with the remote scenario (G1 and G3), while the other half started with the presentational scenario (G2 and G4). Then, the groups switched to complete the levels in the remaining scenario.

In the remote scenario, levels L1 and L2 would be those already mentioned in the previous Chapter 4. For the presentational scenario, these levels would be transformed into their symmetric ones (with a vertical
symmetry line). Figure 5.2 shows the four maps, to address presential and remote scenarios.

![Figure 5.2: Sokoban Levels for Presential and Remote Scenarios.](image)

The investigators randomly assigned L1 to one child per group (for him to be the "Map Explorer") in the remote scenario. Then the remaining attribution of levels would be influenced by this choice. For example, because it was randomly assigned that for the remote scenario, the child that would be the Map Explorer in the first group (G1) was C1, then C2 would be responsible for L2 in this scenario. For the presential scenario, this attribution would be the opposite (C1 would be the Map Explorer for L2 and C2 for L1). Table 5.3 summarises the random distribution of maps between children.

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Starting Scenario</th>
<th>Child ID</th>
<th>Maps Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L0 Remote L1 L2 L2 Pre L1 Pre</td>
</tr>
<tr>
<td>G1</td>
<td>Remote</td>
<td>C1</td>
<td>X X X X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C2</td>
<td>X X X X</td>
</tr>
<tr>
<td>G2</td>
<td>Presential</td>
<td>C3</td>
<td>X X X X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C4</td>
<td>X X X X</td>
</tr>
<tr>
<td>G3</td>
<td>Remote</td>
<td>C5</td>
<td>X X X X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C6</td>
<td>X X X X</td>
</tr>
<tr>
<td>G4</td>
<td>Presential</td>
<td>C7</td>
<td>X X X X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C8</td>
<td>X X X X</td>
</tr>
</tbody>
</table>

Table 5.3: Maps assignment to each pair of children

### 5.5 Data Collection and Analysis

All sessions were audio and video recorded for further analysis.

We performed a statistical analysis to assess the data obtained in the study. The investigators timed how long it took the groups to solve each level. Time started counting when the Map Explorer finished positioning the robot and crate on the map. It was stopped when the crate was pushed by the robot.
to the yellow cap. The investigator also noted the number of “errors” they have made. We defined an “error” as when an iteration has one or more wrong blocks on it (i.e. taken the present position of the robot, it did not take the minor path to reach the objective). An error were only counted from the moment the Block Commander clicked on “Enter”, to send the instructions to the Map Explorer (remote scenario) or to the robot (presential scenario).

Besides, after each session, children were invited to answer a questionnaire (Appendix D) about the activity in general and for both scenarios - remote and presential. From the questionnaire, we also preformed One-way repeated measures Analysis Variance to extract information about their preferences, difficulties and opinions about the system.

From our observation of the videos, we wanted to understand in which scenario (remote or presential) it was more evident that our system supported collaboration and CT learning while promoting an engaging activity.

5.6 Findings

We designed the activities to be engaging while foster collaboration and assist children learning CT. Thus, this section will describe some statistics and observations that we gathered from the study and that helped us understanding children’s opinions about the system and if it was effective meeting their needs.

5.6.1 General Findings

Time (in minutes) for completing L1 and L2 (Figure 5.3(a)): We can see that the duration for solving both levels was higher in the remote context (L1 in remote: \( M = 8,11SD = 5,20 \); L1 in presential: \( M = 3,94SD = 2,20 \), and for L2 in remote: \( M = 13,33SD = 5,11 \); L2 in presential: \( M = 10,06SD = 4,50 \)).

Number of Committed Errors or Mistakes in L1 and L2 (Figure 5.3(b)): On L1, children committed more errors during the remote scenario (L1 in remote: \( M = 1SD = 0,82 \); L1 in presential: \( M = 0,25SD = 0,5 \)). However, for L2, the highest value was for the presential scenario (in remote: \( M = 1SD = 1,16 \); L2 in presential: \( M = 1,5SD = 2,38 \)).

How much Inclusive were the activities for children? (Figure 5.4): Some children believed the system was more inclusive during the presential scenario (\( M = 4,63SD = 0,74 \)) than the remote scenario (\( M = 4,5SD = 0,76 \)).
Time (in minutes) for completing L1 and L2

(b) Number of committed mistakes/errors

**Figure 5.3:** Charts showing duration and the number of errors committed by children on each level, for both scenarios (remote and presential)

**Figure 5.4:** Chart showing children’s opinion about the inclusiveness of our system.

5.6.2 Engagement

During the activities, we saw children laughing when the robot took a wrong direction (and they were not expecting) and bumped into a wall, and sometimes when the robot spoke (at the end of an iteration).

C1 was amazed, perceiving the lights that the robot made while moving. Also, although C1 stated the assembling mechanism was difficult for him because the blocks did not fix the plate at his first attempts, C3 said she liked the blocks and played with them while waiting for new instructions from C4. She pretended they were cookies that went to the oven each time she put them inside the Magic Box.

How much did Children like the Activities? How much fun did they have? How easy they considered the activities? How much did they believe they could do the activities alone? (Figure 5.5): All children said they liked the activities. Four of them said that what they liked the most was being the Map Explorer. Three answered they preferred to be the Block Commander. However, C4 answered that her favourite part was playing with C3. When asked how much fun they had, the results
hardly diverged from the five ($M = 4.75, SD = 0.46$). About the activities’ difficulty, answers were more divergent ($M = 4.25, SD = 0.71$).

**Figure 5.5:** Charts showing children’s answers about the activities: engagement and difficulties.

**How much did Children like the activities specifically for both scenarios?** (Figure 5.6): Children’s responses to this question were not highly different than for the previous general question (of Figure 5.5). However, they have shown their preference for the presentational scenario (remote: $M = 4.63, SD = 0.52$; presentional: $M = 4.88, SD = 0.35$).

**Figure 5.6:** Chart showing children’s responses about how much they enjoyed the activities, for both scenarios.

**How much did Children like to be the Map Explorer and the Block Commander?** (Figure 5.7): In the presentational scenario, children liked to be the Map Explorer, as much as they liked to be the Block Commander ($M = 4,5 SD = 0.53$ for both). However, in the remote scenario they preferred to be the Map Explorer (Map Explorer: $M = 4,63 SD = 0.52$; Block Commander: $M = 4,5 SD = 0.53$).

### 5.6.3 Computational Thinking

During our study, we were observing if children could use a series of Computational Thinking processes, such as: abstraction, algorithm and procedures, debugging, data collection, problem decomposition and
pattern recognition [22].

Thus, similar to previous work [20], we could observe:

- Problem decomposition when children broke larger problems into smaller components;
- Data collection when they gathered relevant information to solve a problem (for example understanding the robot’s and crate positions and counting how many forward instructions were needed to reach the objective);
- Algorithm and procedures when children identified or assembled an ordered set of blocks;
- Debugging, when they identified the output was not the desired one, and tried to correct their instructions.

To reach their objective, children had to analyse the map to gather information about which instructions they should use (so they were using data collection). As explained in Chapter 3, Sokoban belonged to the puzzle category [32], and its complexity relies (among others) on the branching factor. Thus, children had to decompose the solution’s branches into smaller steps and organise them (especially in L2, where the Crate need to reach a specific position before being pushed to the objective floor, meaning these two sequences of commands cannot be inverted). Thus, if children reached the objective, they were using problem decomposition and algorithms and procedures.

All children completed the levels, on the different scenarios.

During the study, we saw some spoken examples of debugging - C2 and C5 (Map Explorers) asked their friends to check the instructions on the plate; and C8 asked C7 for repeating the instructions to see if they matched with what he had assembled.

### 5.6.4 Collaboration

During the remote activities, collaboration was strictly necessary for children to complete the levels. However, we also witnessed collaboration during the presentational scenario activities.
For example, during L2, C3 (the Map Explorer) was confused defining left and right directions. Despite correctly simulating the robot’s movement with her finger, she said the opposite when communicating the instruction to her friend. After two errors, C4 helped her friend by saying the equivalent of the two blocks C3 was confusing while justifying with her finger on the map. After that, C3 stopped confusing the left and right directions and proceeded with the following instructions.

We also asked children’s opinion about collaboration, to understand if they felt the system was collaborative and in which scenario they felt the activities were more collaborative for them.

**How much collaborative were the activities for children?** (Figure 5.8): Children believed collaboration was more present during the remote scenario \( (M = 4.75, SD = 0.46) \) than the presential scenario \( (M = 4, 63SD = 0.52) \).

![Figure 5.8](image)

**How relevant children believed their collaboration were for solving the activities?** (Figure 5.9): Children believed their collaboration was more important during the remote scenario than the presential scenario (remote: \( M = 4.63SD = 0.52 \); presential: \( M = 4, 13SD = 0.64 \)).

![Figure 5.9](image)

When we asked children if they could solve the activities alone (fig. 5.5) \( (M = 4SD = 1.41) \), C1, who had some difficulties assembling the blocks (“This is not helping me”) answered with the lowest value; followed by C7, who during the presential activity (while he was the Map Explorer), started discussing with his friend about the robot’s next steps (after saying the first five right): C7—“To the left?” C8—“going to the left we bump against the wall.”

### 5.7 Discussion

In order to answer our research question: **RQ: How does a remote and inclusive programming learning kit with a robot, tangible blocks, and spatial puzzles support Computational Thinking learning**
and Collaboration compared to a co-located variant?, we tested our kit within both scenarios (remote and presentional).

In general all children liked the robotic coding kit and the activities with the Sokoban game. They said it was fun because they had to program a robot, because they liked solving the maps and assembling the blocks and because it was something new for them, and gave them the opportunity to use the computer. One child also highlighted the blocks were fun because she pretended they were cookies when going to the oven.

Despite C5 has rated the inclusiveness of the system as 3, we believe it was because he felt challenged directing the robot through the right path at L1, on the remote scenario. After the first iteration leaded the robot through a wrong path, and because he was getting quiet and nervous, his SNE gave him some instructions: “Probably its best if you do not spin the map each time you give an instruction to your friend. Try to move the robot instead, because you have to make it take the crate to the marked position.” Then the child tried to follow her advises and as it was working, he was calming himself.

Also, we believe both roles were interesting for them, given the answers about how much they enjoyed playing them, in both scenarios (Figure 5.7). Besides, when asked what they liked the most during the activities, four of them answer being the Map Explorer and three answered they preferred to be the Block Commander.

Through the findings, we understand that all children believed the second level was more complicated than the first one. In fact, their times for completing L2 were superior for both scenarios. Despite the errors being only slightly higher, there were times when a child started by saying the wrong path, but before asking for blocks kept thinking on the solution.

However, all children succeed in completing this (and all) level(s). Thus, because Sokoban addresses a series of CT processes, and because children were capable of debugging their errors in both scenarios, we believe our system supports Computational Thinking.

Finally, children collaborated during all of the activities: not only on the remote scenarios, where their collaboration was imposed by the system, but also for the presentional scenario. For example, C4 started the activities as being the Map Explorer (presentional - L1). However, she did not asked C3 for assembling the blocks she ordered. Instead, she looked at the map and said “C3, I think we should do (…), do you agree?”.

Therefore, we believe that despite the collaboration between the children was equally notorious in both cases (in the remote case it was forced by the assignment of roles to the children), CT was fostered in a more balanced way in the remote scenario, because when children were struggling, the interference from the other player was only verbal and depended on the information they exchanged.
6

Conclusion

Contents

6.1 Achievements ........................................... 61
6.2 System Limitations .................................. 61
6.3 Future Work ........................................... 61
The number and variety of accessible programming kits are increasingly available for visually impaired children. However, most kits do not ensure a balanced distribution of the system’s knowledge between them. Besides, they require children’s physical presence, impeding them to collaborate with their friends while solving CT activities.

6.1 Achievements

We designed an engaging and accessible remote robotic coding kit for children, considering previous work.

Then, we performed user tests with mixed visual abilities children to validate our solution. Our results showed that it supported CT learning and fomented collaboration, and it seemed to balance the systems’ information between children.

Also, we understood some differences between its co-located variant, as collaboration has more naturally arisen on the presentational one, but CT is ensured on the remote scenario.

Finally, children suggested some improvements for our kit.

6.2 System Limitations

We only had four groups of children for conducting our studies. Thus, with a higher number our results would be more reliable, resulting in a more significant analysis of our data and helping avoid errors as testing a possibly atypical sample.

Also, we could identify some complications with our kit - such as the robot needing to be better calibrated, the mechanism for assembling the blocks needed to be redesigned for an easier version and (sometimes) the Zoom meeting quality.

6.3 Future Work

Children suggested some improvements such as the ones from section 6.2. Besides, a child also suggested making larger maps and robots.

We also think a next step could be integrate another CT concepts such as loops or conditionals, making blocks for both.
Bibliography


65


[34] G. Cardoso, A. C. Pires, L. V. Abreu, F. Rocha, and T. Guerreiro, *LEGOWorld: Repurposing Commodity Tools & Technologies to Create an Accessible and Customizable*
Informed Consent Form 1

The "Informed Consent Form 1" is a form that children’s parents had to sign to authorize their participation in our first study.
Consentimento Informado

Programação Remota - Atividades virtuais colaborativas

Somos uma equipa de investigação do Instituto Superior Técnico (Universidade de Lisboa), constituída por: duas alunas de Mestrado em Engenharia Informática e de Computadores (Diana Mendes e Cristiana Antunes); por duas alunas de Doutoramento (Isabel Neto e Filipa Rocha) e pelo Professor Hugo Nicolau.

O objetivo deste trabalho consta na validação de um sistema de programação de atividades virtuais colaborativas, destinado a momentos de lazer, por forma a estimular o pensamento computacional e a fomentar a colaboração entre crianças. Vimos por este meio pedir-lhe autorização para que o sistema possa ser utilizado pelo seu filho, o qual recorre ao uso de um computador e de peças de programação (ou em papel ou disponíveis em Anexo_1.pwp).

A experiência decorrerá durante o mês de Maio, remotamente via Zoom, e terá presente elementos da equipa de investigação por forma a auxiliar/monitorizar a mesma. Peço que leia este documento cuidadosamente e que procure esclarecer quaisquer questões que possa ter. Para tal, poderá utilizar os seguintes contactos:

- E-mail - diana_smm@hotmail.com
- Telefone - 927331254.

Em que consiste o estudo?

Este estudo foca-se no planeamento e desenvolvimento de um sistema remoto de codificação das ações de uma personagem virtual, de modo a que este possa ser utilizado por crianças em momentos de lazer. Este sistema irá estimular o trabalho cooperativo entre duas crianças, sendo que estas devem colaborar para a resolução de uma atividade remota, na qual terão que comunicar via Zoom enquanto reagem aos comportamentos da personagem. Este estudo tem como objetivos:

- averiguar a importância de um kit de programação remota na estimulação do pensamento computacional e da colaboração entre crianças;
- inquirir sobre as suas dificuldades e interesses enquanto estas participam na resolução de uma atividade cooperativa.

O que vamos pedir ao seu filho?

Se autorizar o seu filho a participar neste estudo, e caso esteja confortável com o mesmo no momento, ser-lhe-á pedido que resolva uma atividade em colaboração com outra criança.
A atividade consistirá num jogo de programação de uma personagem virtual, em que as crianças poderão manipular as ações da mesma por meio de blocos de programação. Assim sendo, uma das crianças deverá utilizar peças previamente disponíveis para programar a personagem, enquanto que a outra deverá monitorizar as ações da mesma e negociar com a primeira criança sobre quais seriam os seguintes passos. A atividade estará completa quando a personagem e/ou as peças (dependendo da atividade) estiverem nos locais assinalados no mapa.

A atividade decorrerá na casa de cada criança. Como já referido, a comunicação entre ambas será estabelecida através do Zoom. Para facilitar a recolha dos dados da experiência, a mesma será gravada (áudio e vídeo). No entanto, o anonimato das crianças será sempre garantido pela equipa de investigação associada. Estas gravações poderão ser utilizadas em contexto científico, mas a imagem e voz das crianças serão sempre distorcidas de forma a impedir o seu reconhecimento. Por fim, posteriormente, será realizada uma entrevista às crianças por forma a compreenderem-se dificuldades e a esclarecer, discutir e ampliar as observações produzidas.

**Riscos e benefícios**

Não existe nenhum potencial risco nem benefício para os participantes.

**Confidencialidade dos dados**

Todos os dados captados relativamente à experiência acima referida, serão confidenciais e analisados apenas pela equipa de investigação associada a este projeto. Como referido anteriormente, os dados poderão ser utilizados em contexto científico desde que devidamente anonimizados. O responsável pelo tratamento dos dados será o investigador responsável, o Professor Hugo Nicolau.

Caso necessite de entrar em contacto com o Encarregado de Proteção de Dados da Universidade de Lisboa, poderá fazê-lo através de comunicação escrita dirigida a: Encarregado de Proteção de Dados (DPO, Data Protection Officer) para rgpd@ulisboa.pt. Como responsável pelo participante tem direito a solicitar ao DPO o acesso aos dados pessoais que digam respeito ao seu filho. Tem também os direitos de retificação, remoção, limitação e oposição do tratamento, incluindo o direito de retirar consentimento em qualquer altura, sem prejuízo da licitude do tratamento eventual e previamente consentido. Adicionalmente, tem também o direito de apresentar uma reclamação à Comissão Nacional de Proteção de Dados.

Importa reiterar que a participação do seu filho é voluntária e este poderá sempre optar por não responder ou mesmo desistir a qualquer momento sem qualquer penalização ou consequência.
Declaração de consentimento

Eu, ___________________________________________, encarregado de educação do aluno(a) ___________________________________________, declaro que li a informação acima e que recebi resposta a todas as questões que coloquei. Ao assinar este documento autorizo a participação do meu educando no estudo e consequente gravação.

O Encarregado de Educação

____________________________________________________

Data: ____/____/______

Investigador condutor do estudo

____________________________________________________

Data: ____/____/______

Investigador responsável:

Hugo Nicolau

Professor Auxiliar do Departamento de Eng. Informática do Instituto Superior Técnico, Universidade de Lisboa
Investigador do ITI/LARSyS
http://web.tecnico.ulisboa.pt/hugo.nicolau/
hugo.nicolau@tecnico.ulisboa.pt

Este documento será guardado pelo investigador por pelo menos três anos após o final do estudo
Informed Consent Form 2

The "Informed Consent Form 2" is a form that children’s parents had to sign to authorize their participation in our second study.
Consentimento Informado

Programação Remota e Inclusiva de Robôs

Somos uma equipa de investigação do Instituto Superior Técnico (Universidade de Lisboa), constituída por: duas alunas de Mestrado em Engenharia Informática e de Computadores (Diana Mendes e Cristiana Antunes); por duas alunas de Doutoramento (Isabel Neto e Filipa Rocha) e pelo Professor Hugo Nicolau.

O objetivo deste trabalho é a criação de um sistema de programação de robôs, acessível e inclusivo, destinado a momentos de lazer, por forma a estimular o pensamento computacional e a fomentar a colaboração entre crianças com capacidades visuais mistas. Vimos por este meio pedir-lhe autorização para que o sistema possa ser utilizado pelo seu filho. O sistema consiste num computador ou tablet, peças de programação, um mapa feito (maioritariamente) com peças Lego e um robô de pequenas dimensões.

A experiência decorrerá em outubro, presencialmente (na escola), e terá presente elementos da equipa de investigação por forma a auxiliar/monitorizar a mesma. Peço que leia este documento cuidadosamente e que procure esclarecer quaisquer questões que possa ter. Para tal, poderá utilizar os seguintes contactos:

- E-mail - diana.mendes@tecnico.ulisboa.pt
- Telefone - 927331254.

Em que consiste o estudo?

Este estudo foca-se no planeamento e desenvolvimento de um sistema remoto de codificação das ações de um robô, de modo a que este possa ser utilizado por crianças normovisuais e crianças com deficiência visual. Este sistema irá estimular o trabalho cooperativo entre duas crianças, sendo que estas devem colaborar para a resolução de uma atividade remota, na qual terão que comunicar via Zoom e/ou presencialmente, enquanto reagem aos comportamentos do robot - comportamentos estes que serão acessíveis e inclusivos. Este estudo tem como objetivo averiguar a importância de um kit de programação remota na estimulação do pensamento computacional e da colaboração entre crianças com diferentes capacidades visuais, enquanto estas participam na resolução de uma atividade cooperativa.
O que vamos pedir ao seu filho?

Se autorizar o seu filho a participar neste estudo, e caso este esteja confortável com o mesmo no momento, ser-lhe-á pedido que resolva uma atividade em colaboração com outra criança. A atividade consistirá num jogo de programação de um robô, onde os seus cenários serão construídos em LEGO e as crianças poderão manipular as ações de um robot (Ozobot - elemento dinâmico e programável) por meio de blocos de programação impressos em 3D. Assim sendo, uma das crianças deverá utilizar peças previamente disponíveis para programar o robot, enquanto que a outra deverá monitorizar as ações do mesmo e negociar com a primeira criança sobre quais seriam os seguintes passos. A atividade estará completa quando o robô e/ou as peças (dependendo da atividade) estiverem nos locais assinalados no mapa.

Fig. 1 e 2 - Mapa construído com peças Lego. Chão, caixa e blocos programáveis impressos em 3D. O robot (Ozobot Evo) deverá empurrar a caixa para o local assinalado com um X para o jogo terminar, comandado através das instruções dadas pelo blocos à direita.

A atividade decorrerá na escola das crianças, com o kit previamente desinfetado e disponibilizado a cada uma delas. Como já referido, a comunicação entre ambas será estabelecida através do Zoom e/ou presencialmente. Para facilitar a recolha dos dados da experiência, a mesma será gravada (áudio e vídeo). No entanto, o anonimato das crianças será sempre garantido pela equipa de investigação associada. Estas gravações poderão ser utilizadas em contexto científico, mas a imagem e voz das crianças serão sempre distorcidas de forma a impedir o seu reconhecimento. Por fim, posteriormente, será realizada uma entrevista às crianças, por forma a compreenderem-se dificuldades e a esclarecerem-se, discutirem-se e ampliarem-se as observações produzidas.
**Riscos e benefícios**
Não existe nenhum potencial risco nem benefício para os participantes.

**Confidencialidade dos dados**
Todos os dados captados relativamente à experiência acima referida, serão confidenciais e analisados apenas pela equipa de investigação associada a este projeto. Como referido anteriormente, os dados poderão ser utilizados em contexto científico desde que devidamente anonimizados. O responsável pelo tratamento dos dados será o investigador responsável, o Professor Hugo Nicolau.

Caso necessite de entrar em contacto com o Encarregado de Proteção de Dados da Universidade de Lisboa, poderá fazê-lo através de comunicação escrita dirigida a: Encarregado de Proteção de Dados (DPO, Data Protection Officer) para rgpd@ulisboa.pt. Como responsável pelo participante tem direito a solicitar ao DPO o acesso aos dados pessoais que digam respeito ao seu filho. Tem também os direitos de retificação, remoção, limitação e oposição do tratamento, incluindo o direito de retirar consentimento em qualquer altura, sem prejuízo da licitude do tratamento eventual e previamente consentido. Adicionalmente, tem também o direito de apresentar uma reclamação à Comissão Nacional de Proteção de Dados.

Importa reiterar que a participação do seu filho é voluntária e este poderá sempre optar por não responder ou mesmo desistir a qualquer momento sem qualquer penalização ou consequência.
Declaração de consentimento

Eu, __________________________________________, encarregado de educação do aluno(a) __________________________________________, declaro que li a informação acima e que recebi resposta a todas as questões que coloquei. Ao assinar este documento autorizo a participação do meu educando no estudo e consequente gravação.

O Encarregado de Educação
____________________________________________________
Data: ____/____/______

Investigador condutor do estudo
____________________________________________________
Data: ____/____/______

Investigador responsável:

Hugo Nicolau

Professor Auxiliar do Departamento de Eng. Informática do Instituto Superior Técnico, Universidade de Lisboa
Investigador do ITI/LARSyS
http://web.tecnico.ulisboa.pt/hugo.nicolau/
hugo.nicolau@tecnico.ulisboa.pt

Este documento será guardado pelo investigador por pelo menos três anos após o final do estudo
The "User Study" is a document that we followed to conduct our second study.
USER STUDY SCRIPT

RESEARCH QUESTIONS

1. Será que este kit (robótico) remoto de programação:
   a.Ajuda a desenvolver o Pensamento Computacional?
   b. Ajuda a fomentar a colaboração entre crianças com diferentes capacidades visuais?

2. How does a remote and inclusive programming learning kit with a robot, tangible blocks, and spatial puzzle support learning and collaboration compared to a co-located variant?

PARTICIPANTS

8 participantes - 4 grupos de duas crianças -, em que 4 crianças são normovisuais e 4 crianças têm deficiência visual. Terão entre os dez e os catorze anos. Os participantes serão recrutados através da escola Padre Alberto Neto, Rio de Mouro e Escola Básica de Cinfães.

PREPARATION

Materiais necessários para conduzir o estudo com utilizadores:

Papéis:
- Questionário:
  - a) https://docs.google.com/document/d/1eCm1Sza4z9YoSc6Oxj4SYrK-Y8My7PV13Y2J2_VE/edit
  - Para avaliar o impacto que a atividade teve na colaboração entre as crianças.
  - Para avaliar o impacto que a atividade teve no pensamento computacional das crianças.
  - Para avaliar o impacto entre ambiente presencial VS remoto.
- Pedido de Consentimento

Link: https://thesis-collab.herokuapp.com

Devices:
- 2 Computadores com:
  - Ligação Bluetooth
  - Python, pip, e bleak instalados
  - Firefox ou Chrome - com accessibility
  - Zoom instalado
  - Script “teste.py”
  - Pelo menos 3 portas USB
- 2 ratos USB
- 2 Ozobots
- 2x5 blocos de cada cor - 40 blocos
- 2 Caixas de Blocos com divisórias
- 2 Câmeras Trust
- 2 Caixas da Atividade
- 4 Mapas Lego já montados (seguindo o Guião em Anexo)
- 2 caixas imperiais
- 2 anéis (adereços dos robots para empurrar as caixas)
- 2 Tapa teclas

Controlo e debug:
- Multímetro
- Mini chave

Recording Materials:
- 2 Câmaras
- PC Diana
- Zoom

To Do Entre Sessões:
- Carregar a bateria do robot. Quando estiver totalmente carregado - luzes verdes não intermitentes -, estará pronto para ser utilizado.
- Carregar a bateria do material de gravação.
- Guardar as gravações na Drive - Limpar memória das câmaras.
- Desinfetar peças.

To Do Antes de Cada Sessão:
- Verificar a bateria das câmaras
- Verificar a memória das câmaras
- Ligar as câmaras ao PC e colocá-las no topo das caixas mágicas
  - (Foto)
- Ligar os USBs das lâmpadas e medir a tensão das lâmpadas com o multímetro (entre 2.87 e 2.90)
- Os computadores devem estar ligados à corrente (entre outros, para não causarem variações de tensão à lâmpada)
- Verificar que o Bluetooth dos dois computadores está ligado.
- Colocar tapa teclas
- Ligar Rato USB
- Montar o equipamento de gravação (i.e., 2 microfones e 2 (?) câmaras), de modo a que as câmaras consigam captar o cenário e a interação das crianças e a que os microfones consigam captar o discurso entre as crianças.
- Assegurar que o link começa em https (caso contrário a câmara não irá ligar) (https://thesis-collab.herokuapp.com)
- Testar uma vez o sistema
A caixa está a ler as peças corretamente?
O robot movimenta-se correctamente no mapa?

Ordem da Sessão:

- Início da Sessão:
  - Antes de começar o Nível 0;
  - Nível 0
- Remoto VS Presencial
  - Remoto: Antes de começar o Nível 1;
  - Remoto: Nível 1 e Nível 2
  - Presencial: Antes de começar o Nível 1;
  - Presencial: Nível 1 e Nível 2
- Fim da Sessão

MAPAS: Nível 1 e Nível 2 - Remoto e Presencial

Início da Sessão:

- Antes de começar o Nível 0:

  Investigador:

  - Colocar Script a correr em ambos os PCs
    - Linha de comandos em Downloads -> correr o comando **python teste.py**
    - Não pode ser antes por causa das baterias dos robots.
  - Verificar **conexão ao Robot** (“Am I connected? Yes”).
  - Colocar câmaras e microfones a gravar para ambas as crianças.

- Nível 0:

  Ver script para o Nível 0 (*1)

  Investigador: (*)
Clicar em Esc -> player3.html (HTML da atividade presencial conjunta e do nível 0)

Escolher a câmara: Full HD webcam

Explicar quais são as teclas a usar e as suas funções:
  - “Espaço” regista as instruções
  - (Neste caso o) “Enter” envia a mensagem (diretamente) ao robot

Crianças:

Montar o mapa: Colocar a caixa no quadrado preto; e o Robot no branco (“pusher virado para o corte do círculo)

Remoto VS Presencial:

- **Remoto: Antes de começar o Nível 1:**

  Investigadores:
  - Iniciar Reunião Zoom e colocar a gravar

- **Remoto: Nível 1 e Nível 2:**

  Comandante do Robot:
  - (Ajudar a) escolher a câmara: Full HD webcam
  - Explicar quais as teclas a usar e as suas ações:
    - “Espaço” regista as instruções
    - “Enter” envia a mensagem ao Explorador do Mapa

Criança:

- Clickar no botão azul
- Escolher a câmara: Full HD webcam

Explorador do Mapa:

Investigador:

- Colocar Script a correr no seu PC (abrir uma linha de comandos em Downloads e correr o comando python teste.py)
- Verificar que a conexão ao Robot está correcta (“Am I connected? Yes”)
- Explicar qual tecla usar e as suas ações:
“Enter” envia a mensagem ao robot

**Criança:**

- **Montar o mapa:** Colocar a caixa no quadrado preto; e o Robot no branco (“pusher virado para o corte do círculo)
- **Clickar no botão amarelo**

**PAUSA** - É necessário montarem-se os novos mapas (níveis)

**Investigadores:**

- **Param a reunião Zoom** (caso a atividade remota tenha sido antes da presencial)

- **Presencial: Antes de começar o Nível 1:**

**Investigadores:**

- **Colocar as câmaras e microfones a gravar**

- **Presencial: Nível 1 e Nível 2:**

**Investigador:**

- **Clica em Esc -> player3.html** (HTML da atividade presencial conjunta e do nível 0)
- **Escolher a câmera:** **Full HD webcam**
- **Relembra quais são as teclas a usar e as suas funções:**
  - **“Espaço”** regista as instruções
  - (Neste caso o) **“Enter”** envia a mensagem (diretamente) ao robot

**Criança:**

- **Monta o mapa:** Colocar a caixa no quadrado preto; e o Robot no branco (“pusher virado para o corte do círculo)

**Fim da Sessão:**

- **Entrevistas**

**PROCEDURE**

Explicar aos participantes que se trata de uma atividade colaborativa para a resolução de um jogo de Sokoban.
A atividade terá 2 cenários - presencial e remoto - e um total de 5 níveis, em que o Nível 0 será igual para ambas as crianças e passado individualmente. Os restantes níveis serão jogados em conjunto. Dependendo do número da experiência, as crianças poderão começar a atividade com o cenário remoto ou presencial. Em ambos os casos, resolverão 2 níveis (Nível 1 e Nível 2).

No cenário remoto (e no nível 1), os investigadores atribuir-lhesão papéis - Comandante do Robot ou Explorador do Mapa. Aquando o segundo nível irão trocar os mesmos.

Para escolher o seu papel, a criança deverá selecioná-lo no Menu inicial:

- O botão azul à esquerda, com a imagem dos blocos, fará da criança a “**Comandante do Robot**”.
- O botão amarelo à direita, com a imagem do robot/mapa, fará da criança a “**Exploradora do Mapa**”.

**Nível 0:**

**[Para que serve?]**

Tem o propósito de fazer as crianças:

- **Compreenderem o objetivo do jogo** - i.e., colocarem a caixa no local assinalado no mapa (peça de cór amarela e com uma cruz)
- **Perceberem como poderão manipular o robot** para completarem o objectivo (utilizando os blocos).

Assim sendo, as crianças deverão:

- Pegar nos **blocos de instruções e simular o movimento do robot**, para que este empurre a caixa até à zona assinalada.
- Conseguir **explorar noção de unidade** (onde começa e termina uma casa).
- Conseguir **identificar as 3 casas diferenciadas** do resto do mapa:
  - Amarela - Objetivo
  - Preta - Onde começa a caixa
  - Branca - Onde começa o robot

**Script**

*Investigadores apresentam-se individualmente* - nesta altura, as crianças encontram-se juntas - e explicam-lhes:
- Hoje irão realizar uma atividade colaborativa do jogo Sokoban, já ouviram falar?
  *Pauses*

- É um jogo antigo que foi adaptado para esta atividade em específico, e para perceberem melhor os objetivos deste jogo, decidimos preparar um pequeno nível para vocês poderem explorar à vontade como funciona

- Portanto, por agora vamos separar-vos, para ambos terem a oportunidade de explorar o jogo à vontade.

  *Separar crianças* - diria em salas diferentes para ambas se concentrarem e não ouvirem os comentários uma da outra.

- Então, este é o Nível 0.

- Aqui está o mapa pelo qual te vais guiar. As peças deste mapa têm cores e texturas diferentes para serem mais facilmente distinguídas

  *Mostrar à criança - devem ver e sentir as diferenças entre as peças*

  - A peça branca com um círculo cortado marca a posição de partida do robot - o corte deverá estar na mesma direção que o “pusher” do robot;
  - A peça preta com um quadrado marca a posição inicial da caixa;
  - A peça amarela com uma cruz marca o objetivo - o local para onde o robot deverá empurrar a caixa.

- E estas são as peças que conduzem o robot. Uma vez mais também contém cores e relevos diferentes.

  *Mostrar à criança - devem ver e sentir as diferenças entre as peças*

  - A peça amarela com uma seta a apontar em frente indica que o robot **andará para a frente**.
  - A peça vermelha com uma seta a apontar para trás indica que o robot **andará para trás**.
  - A peça azul com uma seta a apontar para a direita indica que o robot **rodará (sem andar)** para a **direita**.
  - A peça verde com uma seta a apontar para a esquerda indica que o robot **rodará (sem andar)** para a **esquerda**.

- Para fazeres o robot andar, precisas de colocar estas as peças neste tabuleiro.

  *Mostrar à criança e deixá-la sentir enquanto explica o seguinte:*

- Para cada peça, o tabuleiro contém **3 pinos**, tal como as peças por baixo contém **3 encaixes**.

  *Virar uma peça, mostrar, dar-lha para a mão e continuar*
- Para saberes se estás a colocar a peça na posição correcta, tens/deves confirmar 2 coisas:
  - A saliência do tabuleiro que deverá estar em baixo, e não em cima.
    *Apontar e fazê-la tocar na saliência*
  - E se reparares, os blocos contém 2 pistas, uma de cada lado. Então a parte saída do bloco deverá estar para a direita.
    *Apontar e fazê-la tocar nas entrada e saída do bloco*

- Então se quiseres fazer o robot andar em frente, por exemplo, o que deverás fazer?
  *Ajudar a responder caso seja necessário. Evitar frustrações*

- Exatamente, colocar uma peça amarela na primeira posição do tabuleiro, que é na primeira linha, a mais à esquerda
  *Colocar peça, caso criança não o tenha feito*

- Então agora vamos ver o robot a andar!
  *Investigador abre o link no PC, clicka no esc e abre a câmara “Webcam full HD”*

- Podes colocar o tabuleiro na caixa mágica, entre os carris.
- Vamos fechar bem a caixa.
- E por fim, para registares as instruções, terás de carregar no espaço.
  *Investigador aponta para o espaço ou puxa a mão da criança para a tecla, caso necessário*

- Para as enviares ao robot, terás de carregar no Enter.
  *Investigador aponta para o Enter ou puxa a mão da criança para a tecla, caso necessário*

*Ambos assistem ao robot a mexer-se*

- Percebeste?
  - *Sim* - Vê lá se consegues resolver o resto do caminho. Qualquer coisa, eu estou aqui e também te ajudo e esclareço qualquer dúvida.
  - *Não* - Então vamos fazê-lo mexer-se mais uma vez sozinho, o que acha? E se tiveres dúvidas vais-mas colocando.

*A resolução do nível segue assim por diante*

**REMOOTO - Nível 1 e 2:**

*Para que serve?*

Tem o propósito de se conseguir responder às perguntas de investigação.

Assim sendo, as crianças deverão:
- **Colaborar** entre si para conseguirem completar o objetivo do jogo (fazerem com que o robot empurre a caixa para a peça amarela com uma cruz).
- Explorar os movimentos do Robot e fazer **sequências** de instruções.
- Conseguir fazer **debug** às instruções, caso seja necessário

**Script**

**Para o Nível 1:**

*Investigadores preparam a sala zoom* - nesta altura, as crianças encontram-se **separadas**.

*Index.html*

*Investigador 1 explica*:

- Desta vez, vocês irão jogar em conjunto, portanto, deverão colaborar para chegar ao objetivo. Nós iremos atribuir-vos papéis diferentes, o que significa que 1 de vós tem acesso ao mapa; o outro tem acesso aos blocos de instrução que controlam o robot.

*Investigador 1 escolhe*:

- Portanto tu [Criança A] serás o Comandante do Robot e terás acesso aos blocos de instrução. Assim sendo, deverás escolher o botão azul da aplicação.
- Tu [Criança B] serás o Explorador do Mapa e terás acesso ao Mapa e ao Robot em si. Por isso, deverás escolher o botão amarelo da aplicação.

*Investigador ajuda na escolha da Câmara no caso do Comandante do Robot*:

*Por fim os investigadores recordam as teclas que as crianças devem usar*:

*Para o Comandante do Robot: **Espaço** (regista instruções) e **Enter** (envia informação ao outro jogador)*

- Agora nós vamos-nos calar e esperamos que se dirvertam a completar este nível.

**Para o Nível 2:**

*De volta ao menu principal - index.html*


*Investigadores podem ajudar a escolher os roles na aplicação*

*Investigador ajuda na escolha da Câmara no caso do Comandante do Robot*:

- E podem jogar (:)

**Presencial - Nível 1 e 2:**

[**Para que serve?**] Tem o propósito de se conseguir responder às perguntas de investigação.
Assim sendo, as crianças deverão:

- **Colaborar** entre si para conseguirem completar o objetivo do jogo (fazerem com que o robot empurre a caixa para a peça amarela com uma cruz).
- Explorar os movimentos do Robot e fazer **sequências** de instruções.
- Conseguir fazer **debug** às instruções, caso seja necessário.

**Script**

**Para o Nível 1:**

*Investigadores preparam os novos mapas* - nesta altura, as crianças encontram-se **juntas**.

*Index.html*  
*Investigador abre o link no PC, clicka no esc e abre a câmara “Webcam full HD”*

- Desta vez, vocês irão jogar em conjunto, portanto, deverão colaborar para chegar ao objetivo.
- Tal com no modo remoto, neste modo presencial também vos serão atribuídos papéis (Comandante do Robot e Explorador do Mapa). Para além disso, o objetivo mantém-se o mesmo (colocar a caixa sobre a peça amarela).

**Para o Nível 2:**

*Investigador troca de mapa*

- Agora têm de resolver este nível e as regras são as mesmas do nível anterior.
- E podem jogar (:)

**APPARATUS**

Describe relevant materials that will be used in the study.

Irão ser utilizados 2 computadores que corram o Windows 10, com pelo menos 2 portas USB e que consigam estabelecer ligações por bluetooth. Os pcs têm que ter o python, pip, bleak e audioSimple corretamente instalados/configurados (variáveis de ambiente), assim como o programa Zoom e o browser Firefox e/ou Chrome (**com o plugin de acessibilidade**). Para além disso, os computadores deverão ter o ficheiro “**teste.py**” na pasta das transferências.

Em adição, serão também utilizados 2 robots Ozobot Evo (com 1 “pusher” cada), 2 “caixas mágicas” (para leitura das peças), 4 mapas LEGO previamente montados e 2 capas para as teclas não utilizadas nos laptops. Serão ainda necessários 40 blocos de instrução (20 para cada criança) e duas caixas com divisórias para melhor arrumação/separação.

Por fim, irão ser utilizados diferentes cenários (4 mapas lego), com com peças impressas em 3D que servirão de caminho para o robot andar. Estas peças contêm diferentes relevos/cores para que as suas funções sejam fáceis de diferenciar. O mesmo se aplica para os blocos de instrução anteriormente referidos.

As experiências serão gravadas por Zoom e por câmaras externas, de forma a melhor se observar a interação que as crianças têm com o protótipo e entre elas.
The “Questionnaire” is a document that we gave children during our second study to understand their preferences about the activities.
QUESTIONÁRIO/ENTREVISTA:

Nome: 
Idade: 
Ano de escolaridade: 
Tipo de deficiência visual e grau: 
Tecnologia assistiva: 

Alguma vez programaste? Lembras-te que tipo de programação é que foi? (linguagem ou estilo Minecraft e por blocos) 

Alguma vez programaste algum robot? Lembras-te do(s) nome(s) do(s) robot(s)? Que tipo de Programação é que usaste para programar o(s) robot(s)? (Blocos, C, etc…)

Atividade Geral

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quão pouco/muito gostaste da atividade?</td>
<td>Gostei mais:</td>
<td>Gostei menos:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quão aborrecida/divertida foi a atividade?</td>
<td>Porque:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quão difícil/fácil foi a atividade?</td>
<td>O mais difícil foi:</td>
<td>O mais fácil foi:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quão pouco/muito acreditas que conseguirias fazer a atividade sozinho?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

· Como é que nós poderíamos melhorar a atividade/jogo?
Atividade Presencial VS Remota

- O que é que fizeram de diferente de um modo de jogo para a outro?

<table>
<thead>
<tr>
<th>Modo de Jogo: <strong>Presencial</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de fazer a atividade presencialmente?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de ser o Explorador do Mapa?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de ser o Comandante do Robot?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito <strong>colaborativa</strong> foi a atividade?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito <strong>inclusiva</strong> foi a atividade?</strong></td>
</tr>
<tr>
<td><strong>Quão relevante foi a tua colaboração na resolução da atividade (de modo geral para ambos os níveis)?</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modo de Jogo: <strong>Remoto</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de fazer a atividade remotamente?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de ser o Explorador do Mapa?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito gostaste de ser o Comandante do Robot?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito <strong>colaborativa</strong> foi a atividade?</strong></td>
</tr>
<tr>
<td><strong>Quão pouco/muito <strong>inclusiva</strong> foi a atividade?</strong></td>
</tr>
<tr>
<td><strong>Quão relevante foi a tua colaboração na resolução da atividade (de modo geral para ambos os níveis)?</strong></td>
</tr>
</tbody>
</table>