LEGOWorld: Repurposing Commodity Tools & Technologies to Create an Accessible and Customizable Programming Environment

GONÇALO CARDOSO, LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
ANA CRISTINA PIRES, LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
LÚCIA VERÔNICA ABREU, LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
FILIPA ROCHA, LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
TIAGO GUERREIRO, LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal

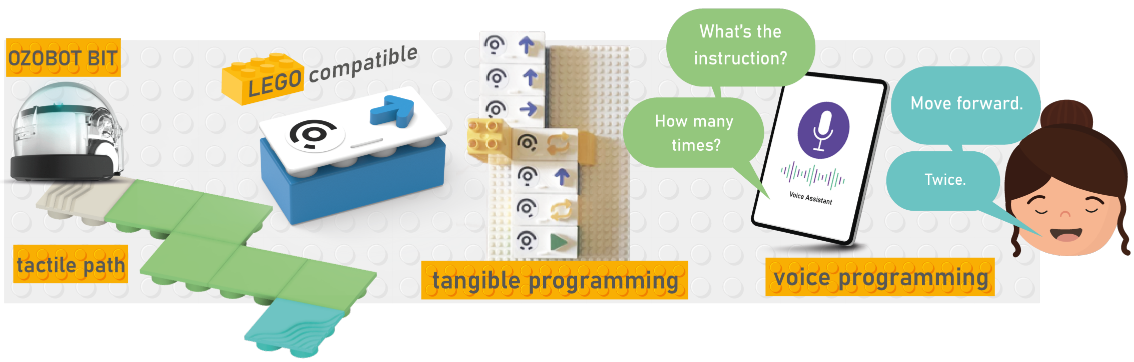

Fig. 1. LEGOWorld. A small-scale robot travels over a path of 3D-printed LEGO caps, assembled over a LEGO base plate. Two interaction modes are represented: Tangible Programming- LEGO blocks with 3D printed programming caps are assembled to create a program, recognized by a mobile device camera; and Voice Programming- programs are created conversationally in a dialog.

We explore how commodity objects and technologies can be repurposed to provide a multimodal programming environment that is accessible to children with visual impairments, flexible, and scalable to a variety of programming challenges. Our approach resorts to four main components: 1) a LEGO base plate where LEGO blocks can be assembled to create maps, which is flexible and robust for tactile recognition; 2) a tangible programming area where LEGOs, with 3D printed caps enriched with tactile icons, can be assembled to create a program; 3) alternatively, the program can be created through a voice dialogue; and 4) a low-cost *OzoBot Bit*. A preliminary study with educators suggests that the approach could be useful to a variety of developmental stages, is accessible and stimulating, and promising for CT training.

CCS Concepts: • **Human-centered computing → Accessibility**.

Additional Key Words and Phrases: inclusive programming, tangibles, voice, LEGO, multimodal, children with visual impairments

ACM Reference Format:
Gonçalo Cardoso, Ana Cristina Pires, Lúcia Verónica Abreu, Filipa Rocha, and Tiago Guerreiro. 2021. LEGOWorld: Repurposing Commodity Tools & Technologies to Create an Accessible and Customizable Programming Environment. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '21 Extended Abstracts), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3411763.3451710

1 INTRODUCTION

Computational Thinking (CT) is becoming a fundamental literacy skill, such as reading and writing [16]. Learning and understanding the application of computational concepts in early childhood promises children the opportunity to foster the development of creativity and cognitive abilities, such as planning, and problem-solving. By learning new methodologies, students change perspectives about their surroundings and how to approach new problems [14, 15, 17]. For children to be able to succeed in a future computational driven society, we should ensure a common and solid basis of computational concepts in every stage of their education [16].

In recent years, CT emerged as a discipline in schools. These concepts are borrowed from computer science, such as sequences and iterations, as well as practices like debugging, abstracting, and reusing [4]. Promoting CT has been carried out mainly through programming environments. These approaches are highly visual and many make use of block-based coding, like Scratch [10] or Blockly [5].

We introduce LEGOWorld (Figure 1), a multimodal system to teach computational concepts to children with visual impairments through tangible or voice input programming. The output is a robot moving on a user-personalised LEGO path. The programming blocks are composed of LEGO blocks and 3D printed caps with the action's icon (e.g., an arrow pointing forward). This system creates repurposability and a play opportunity for children familiarized with LEGO. Also, it provides alternative ways to program the robot, that accommodate diverse ages and developmental stages.

We performed an online study with educators (special needs educators and TI instructors) where they were invited to answer a structured survey, after watching a video demonstrating the features of the approach. Responses suggest that a LEGO-based environment would be welcomed and easily introduced in teaching activities. Also, educators welcomed the flexibility of the programming tasks, and the tactile richness of the maps and blocks.

2 RELATED WORK

Training CT with children has been mainly performed with block-based programming environments. Block-based environments allow the composition of blocks to create code reducing the cognitive load presented in text-based settings. Scratch [10], for example, is widely used in schools for children to learn computational concepts and produce a graphical output. These screen-based virtual environments are visually demanding and not accessible to children with visual impairments. Previous work has tried to bridge this gap by increasing the accessibility of the virtual programming environments (e.g., Blocks4All [12]). Other projects have focused on accessible tangible solutions with audio output [9, 13]. Below, we provide an overview of the main concepts borrowed by our approach towards an inclusive programming environment: robots, tangible user interfaces, and voice user interfaces.

Programming Robots. Robots are available, affordable and engaging to children,and their use is widely popular in the context of programming. There have been efforts to provide children with visual impairments the possibility to interact with robots in the context of programming [1, 12, 14]. Results are promising; robots can be a motivating artifact to reinforce spatial relations and mappings in a programming environment [12, 14].

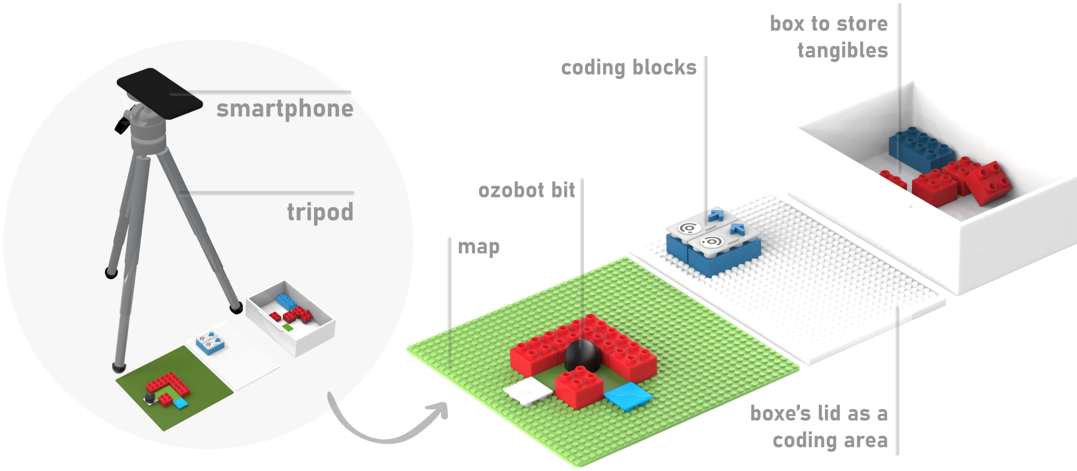

Fig. 2. System Overview with smartphone, tripod, physical map, tangible blocks, and a box with a lid.

Programming with Tangibles. The use of tangible blocks allows children to tinker and reduce cognitive load while simultaneously developing motor, perceptual and cognitive abilities [2]. Despite the potential for being more accessible by design, commercially-available tangible programming environments have failed to accommodate the needs of children with visual impairments [14]. Although physical, they generally miss the tactile cues for action recognition, orientation, and correct assembly. Conversely, recent work has explored how to make this type of environment more accessible with encouraging examples like Torino [13], StoryBlocks [9], or P-cube [8].

Programming with Voice. The emergence of Voice User Interfaces (VUI) in our daily lives present new possibilities in several domains, including education. Children are becoming accustomed to use voice to talk with these interfaces (e.g., Siri) which provides an opportunity to take advantage of the children natural's input. TurtleTalk is a promising example of a conversational programming environment where children create a program in a dialogue with the system. The approach provides an environment where children are supported step-by-step while solving a programming challenge. Still, it relies heavily on visual information to understand the goal and map status [7]. However, there are few studies with VUI and children with visual impairments. Metatla et la. [11] performed studies with educators and mixed-visual ability students demonstrating that VUIs contribute to an inclusive learning experience.

In our approach, we explore the aforementioned qualities and promise of robots, tangibles and voice in an environment built to be accessible, flexible, scalable, and sustainable.

3 CREATING A MULTIMODAL PROGRAMMING ENVIRONMENT WITH LEGO AND VOICE INPUT

We designed and implemented *LEGOWorld* to target the learning of programming concepts and the development of computational thinking in children with visual impairments. To learn these concepts, children play a game with several levels with the help of an assistant that describe the adventures of the protagonist, a robot named *Ozobot Bit*[1] and allows feedback of the given instructions. The system is composed of (a) a physical LEGO map where the robot moves; (b) tangible LEGO blocks to compose a sequence of instructions for the robot movements; (c) a mobile application, and (d) a small robot (Figure 2).

[1]ozobot.com

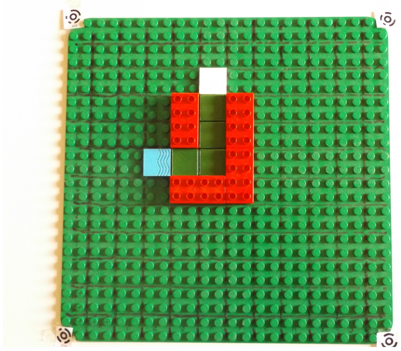
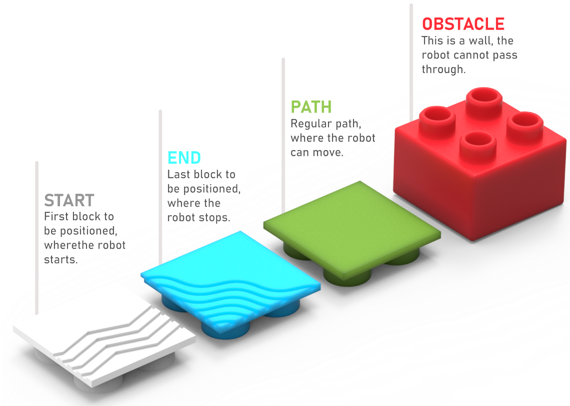

Fig. 3. Tangible Programming Environment. (Left) Physical Map (green base plate) with LEGO and 3D caps showing a green path, red walls and a white start and blue end piece; (Right) LEGO and 3D *Path* caps showing wall, pieces, and start, path and end caps.

LEGO is a central component of our design. We decided to use commodity objects such as LEGO blocks due to their availability and presence in children's environments and their association with playful experiences. It is a commonly used toy with a range of desirable properties, such as a graspable dimension, material resistance, modularity and high contrast colors, making it suitable for an accessible environment. Another advantage of reusing LEGO to build a programming environment is the cost reduction of the material and time needed to fabricate customized 3D printed artifacts from scratch, with specific properties for blind children. Instead, we designed and printed 3D printed accessible caps, compatible with LEGO DUPLO blocks. The repurposability of an existent toy creates an opportunity for making tangible and accessible coding blocks more reproducible and available.

3.1 Tangible Programming Environment

The tangible environment is composed of a LEGO plate and blocks and path LEGO-alike caps. The robot moves in a path of 3D-printed LEGO caps surrounded by LEGO blocks assembled over a LEGO base plate (Figure 3).

Lego Map. We used a green base plate as the world where the robot will navigate. The plate contains Topcodes [6] at the corners to be detected by the camera of the Android device, to facilitate the world and path detection algorithm. LEGO red blocks are used as obstacles or "walls", defining the structure of the path. Other colors could be parameterized.

Path Caps. We developed and printed 3D Path caps to have rich tactile properties and different colors to facilitate their discrimination by touch and/or vision. The white *start cap* has horizontal lines, the blue *end cap* oblique wavy lines and the green *path cap* has no texture.

3.2 Interaction Modes

The system has two distinct interaction modes to move the robot: the tangible mode in which programming blocks are assembled adjacently in order to build the sequence of instructions; and the voice mode in which the user indicates verbally each instruction step by step. In both interaction modes there is a voice assistant in the device's app, that helps the user throughout the game.

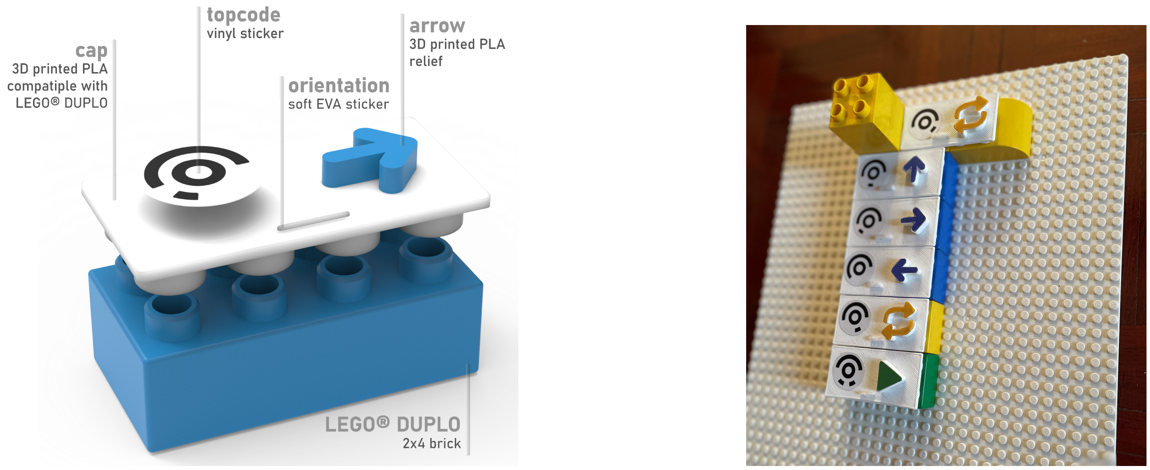

Fig. 4. Programming Blocks. (Left) Breakdown of a programming block. A LEGO with a 3D-printed cap with a Topcode and a tangible turn-right arrow; (Right) Programming Blocks horizontally arranged over the base plate.

Programming with Blocks The programming blocks allow the creation of sequences, to be executed by the robot, by placing them horizontally from the top to the bottom, over a LEGO white plate which serves as a platform to build the code (Figure 4). The blocks have 3D patterned reliefs affording children with visual impairments to identify the block's corresponding function: forward; left; right; start of loop; end of cycle, or play. The movement blocks have blue embossed arrows, the play block has a green embossed triangle typically associated with the function start, and the loop blocks have two yellow embossed arrows, giving the idea of a cycle/repetition.

The 3D patterned reliefs can be fitted to any LEGO, in order to allow the user to play with the available LEGO. Blocks also have Topcodes [6] to facilitate the app detection.

Programming with Voice Alternatively, the system allows a voice interaction mode enabling children to program sequences using their voice (Figure 1). To guide this interaction, the app asks specific questions, step by step. Firstly, the app asks to which direction should the robot move ("forward", "left", or "right"). Secondly, it asks how many times should the robot move in such direction. To end the sequence the user should say "terminate".

3.3 Robot

We decided to use *Ozobot Bit* because of its small dimensions that makes it suitable to a LEGO environment and its low-cost . The robot has a sensor to identify a set of colors and moves accordingly, i.e., each color represents a specific instruction of movement.To load the sequence of instructions the user places the robot over the touchscreen, which then quickly flashes the color codes to the robot.

4 PRELIMINARY STUDY

We sought to understand the possibilities and limitations of the designed approach. To this end, we conducted a survey with educators, seeking to understand their perceptions about the accessibility of the environments, and its components, and how it could foster blind children's inclusion and development.

Participants. Six educators participated in the study, of which 3 are robotics teachers (RT) and 3 special needs educators (SNE). All worked regularly at schools with children, and 4 of them had experience with children with visual impairments (SNE1, SNE2, SNE3 and RT3).

Procedure. The survey was sent by email to schools and institutions of reference for children with visual impairments. Before starting the survey, participants read the informed consent and agreed to participate. The survey started with a video where the research team demonstrated the use of the system and its different components. The survey was composed of 4-5 questions for each section: (1) Educators' experience with children with visual impairments as well with robotics; (2) system's overview; (3) tangible LEGO map; (4) programming blocks; (5) programming with voice; and (6) robot. All sections about each component had questions about the qualities, flaws, opportunities, and educational activities and acquisitions promoted. The responses provided were subject to a thematic analysis by two of the researchers [3].

5 RESULTS

We report the themes that resulted from our inductive thematic analyses.

5.1 LEGOworld learning and cognitive benefits

In general, educators welcomed the system and highlighted its intuitive understanding, ease of use and accessibility. Educators associated the use of the system as a possible tool to: *"stimulate cognitive skills, as memory and reasoning [...], laterality* [2] *concepts and orientation"* -SNE3- as well as children's *"psycho-motor development"* -SNE1, SNE2. Educators remarked that LEGOworld would be very useful for disciplines such as robotics, ICT or in mathematics, and that it could foster the learning of "programming" -SNE2, "basic computational concepts, laterality" - RT1, and "logical reasoning and problem-solving skills" -RT3.

5.2 Children's Age

Two SNEs mentioned that the system could be used by young children, starting at 4-6 or 7-9 years old *"if children have already acquired some laterality concepts" -SNE3* and because *"instructions are easy with immediate feedback"* -SNE2. However, the majority (SNE1, RT1, RT2 and RT3) pointed that it would be ideal for children aged 9-10 years because they *"are very interested in this type of systems and already have the skills to understand and retain the concepts taught so that they can actually learn to use this system autonomously" - RT1.* Also, they *"have a better understanding of how the game works and the logic behind programming"* -RT2. RT3 highlighted that, at the age of 10, educators *"introduce block programming .. [so] this system will work as a fundamental tool to monitor and develop the skills needed at the beginning of programming".*

5.3 LEGOWorld usage at home and school

Educators foresaw the system being used in classrooms: *"I think that today, it would be more practical to use the system in a classroom with a teacher capable of assisting children"* - RT1 - and could also include didactic games or "research projects". It could be used by dividing children in groups, so they have to collaborate to solve the exercises or, as RT1 mentioned, it could also be used individually: *"Each child could have their own LEGO board and blocks, and using robot racing games / exercises, puzzles, or mazes, children could use the blocks as instructions".* SNE3 called attention to the

[2] term used by educators to refer to spatial orientation; notion of sides

fact that sometimes the chaotic environment in classrooms, may impair the recognition of the children's voice: *"In the classroom can be more complicated due to the crossing of sounds"* -SNE3.

Educators also envisioned the use of the system at children's homes. It could be integrated in children's *"daily activities (DA)"*, or it could be used *"collaboratively with siblings as a game"* -RT2- or as a *"playful therapeutic activity"* -SN3. RT3 mentioned that, to be used at home, the system should include explanatory audios with examples of activities. Similarly, RT1 suggested: *"[include] a website with exercises and games with activities to guide parents."*

5.4 Tangible LEGO Environment

Educators were enthusiastic about the tangible LEGO environment and saw its benefits for children with visual impairments, such as: *"easy to assemble and playful because its LEGO"* -RT2. They valued the fact that its physicality allows to have all the tactile information available to be accessed anytime and affords an *"easy understanding of the path and directions"* -RT2.

Educators were excited with the possibility to use LEGO in programming activities: *"Excellent, something familiar and fun"* -RT2 and with its advantages, as *"LEGO blocks are always associated with something playful, so it can be more attractive to trigger children's interest"* -SNE3.

They foresaw the use of the tangible environment to train *"sequences, time, space and succession"* -SNE2, *"initial programming concepts (basic and repetitive movements), to trigger other activities, such as music, or quizz type"* -RT3.

SNE3 indicated that the environment affords the development of *"[tactile] sensitivity, orientation and memory"* and that it could be used to train Orientation & Mobility skills, such as training *"routes that the children need to learn in their day-to-day lives, that can be simulated. Short routes only."*

One educator suggested *"to add more than one board to enlarge the environment."* -RT1

5.5 Tangible and Voice Interaction Modes

The two modes of interaction were also pointed as a great quality of the system. On one hand, *"the use of voice to command the robot and the obtained [audio] feedback"* -RT1 could be very advantageous to children. On the other hand, the use of LEGO were appreciated due to the fact that "they are common objects which leads to an easy access" -RT3, and allow a rich tactile interaction with the robotic LEGO environment. On the other hand, *"recording the voice and verifying the instructions works quite well"* -RT1.

Programming with Tangible LEGO Blocks. Educators mentioned that it was *"a great solution [because LEGO] are not only known by the vast majority of children, but also constitute a great solution for children with vision loss or blind"* -RT1. They considered it to be *"simple and to enable an immediate identification [of its functions]"* -SNE2, and a *"good solution for basic sequences"* -RT3. They *"follow the logic of the sequence of actions, making it simple to debug"* -RT3. Another advantage is their permanence, i.e. tangible blocks are available anytime and children can access that information without having to recruit memory resources. It is *"simple, the orders remain noticeable, the sequence remains available and can be consulted"* -SNE2. However, their simplicity may imply a limitation: *"it only allows basic sequences, as such, it limits programming"* -RT3.

Programming with Voice. The voice interaction mode was considered easier to understand, affording more autonomy and *"less complexity, so it can be more accessible to younger people"* -RT2 and valued as *"a good form of communication for blind children, as it makes them more focused on the activity"* -SNE3. However, young children's pronunciation could difficult the recognition of their words: *"It can be difficult for the system to recognize their commands which could slow the interaction"* -RT1. However, they called to attention that the child needs to already have some *“notions of programming, [so] voice programming will be more effective and faster. [If not, the] system may be more suitable for older children (12-15 years old), as they follow audio instructions easier and formulate more complex operations”* -RT3.

SNE3 recommended to use the system in *“a place with silence with no other voices to avoid confusion"* -SNE3. One of the solutions indicated was to use *“headphones via Bluetooth [in] a classroom context "* -SNE3.

Educators recommended to make this mode of interaction faster: *“the system could be faster to process information, as well as verifying instructions”* -RT1.

5.6 Robot

Educators liked the robot, its dimensions and the possibility to use in a LEGO environment: *“[the robot has the] right size and weight to be manipulated by children”* -SNE2. RT1 mentioned that the*“feedback given is clear and well done ”*. Finally, they suggested to consider music as another type of feedback.

6 CONCLUSION AND FUTURE WORK

We present LEGOworld, an accessible and customizable programming environment for children with visual impairments, that repurposes the use of commodity tools and technology, such as LEGO and Ozobot Bit. We report the results from a online survey with educators (SNEs and RTs) that explored the qualities and limitations of the system, its possible benefits in children’s learning and developmental stages and its appropriateness for programming activities.

Repurposability and Play of LEGO. The familiarity and playful experiences associated with LEGO blocks may facilitate children’s engagement with programming. Educators mentioned that the ease for children to place the elements on the LEGO plate and recombine these blocks in the case of debugging, creates a way to play with the concepts of programming that is engaging, and accessible.

Accessibility of the Programming Tangible LEGO environment. The tangible environment made by using LEGO and 3D caps was very well received due to its rich tactile information and visual colors, making it intuitively accessible to children with visual impairments. The continuous availability of tactile information was also considered of great relevance for children with visual impairments. Educators envisioned that the physicality of the LEGO map and of the robot would allow the child to detect the robot’s movements and changes in direction, which would facilitate debugging and the training of laterality concepts and spatial orientation. They found the system appropriate for learning programming in the context of a classroom as well as at children’s home, suggesting that, at home, it would be necessary a guide for parents with instructions and activities.

Programming with Tangibles Blocks and with Voice Commands. Educators foresaw the opportunities both interaction modes could have in children’s programming experience and learning. On one hand, the tangible blocks allow children to have access to the sequence of instructions anytime, whereas with voice commands, children do not have access to all the instructions. Although programming with voice commands triggers the app’s feedback at each step of the code, with tangibles children would only have the feedback at the end of the sequence of instructions. Educators also mentioned that it is easier for young children to verbalize instructions and to have the feedback step by step. However, the lack of having available all the sequence of instructions when creating the algorithm could complicate their understanding of the actual position or in debugging, or simply imply more memory resources allocated to remember all the instructions.

ACKNOWLEDGMENTS

We kindly thank all educators that collaborated with us. This work was supported by FCT through LASIGE Research Unit funding, ref. UIDB/00408/2020 and ref. UIDP/00408/2020.

REFERENCES

[1] Lúcia Abreu, Ana Cristina Pires, and Tiago Guerreiro. 2020. TACTOPI: A Playful Approach to Promote Computational Thinking for Visually Impaired Children. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, Greece) *(ASSETS '20)*. Association for Computing Machinery, New York, NY, USA, Article 82, 3 pages. https://doi.org/10.1145/3373625.3418003

[2] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. 2017. Learnable Programming: Blocks and Beyond. *Commun. ACM* 60, 6 (May 2017), 72–80. https://doi.org/10.1145/3015455

[3] Richard E Boyatzis. 1998. *Transforming qualitative information: Thematic analysis and code development.* sage.

[4] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *In AERA 2012*.

[5] N Fraser. 2015. Ten things we've learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. 49–50. https://doi.org/10.1109/BLOCKS.2015.7369000

[6] Michael Horn. [n.d.]. Tangible Object Placement Codes. http://users.eecs.northwestern.edu/~mhorn/topcodes/.

[7] Hyunhoon Jung, Hee Jae Kim, Seongeun So, Jinjoong Kim, and Changhoon Oh. 2019. TurtleTalk: an educational programming game for children with voice user interface. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[8] Shun Kakehashi, Tatsuo Motoyoshi, Kenfichi Koyanagi, Toru Ohshima, and Hiroshi Kawakami. 2013. P-cube: Block type programming tool for visual impairments. In *2013 Conference on Technologies and Applications of Artificial Intelligence*. IEEE, 294–299.

[9] Varsha Koushik, Darren Guinness, and Shaun K. Kane. 2019. StoryBlocks: A Tangible Programming Game To Create Accessible Audio Stories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 492, 12 pages. https://doi.org/10.1145/3290605.3300722

[10] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The Scratch Programming Language and Environment. *ACM Trans. Comput. Educ.* 10, 4, Article 16 (Nov. 2010), 15 pages. https://doi.org/10.1145/1868358.1868363

[11] Oussama Metatla, Alison Oldfield, Taimur Ahmed, Antonis Vafeas, and Sunny Miglani. 2019. Voice User Interfaces in Schools: Co-Designing for Inclusion with Visually-Impaired and Sighted Pupils. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300608

[12] Lauren R Milne and Richard E Ladner. 2018. Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), 69:1—-69:10. https://doi.org/10.1145/3173574.3173643

[13] Cecily Morrison, Nicolas Villar, Anja Thieme, Zahra Ashktorab, Eloise Taysom, Oscar Salandin, Daniel Cletheroe, Greg Saul, Alan F Blackwell, Darren Edge, Martin Grayson, and Haiyan Zhang. 2020. Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities. *Human–Computer Interaction* 35, 3 (2020), 191–239. https://doi.org/10.1080/07370024.2018.1512413

[14] Ana Cristina Pires, Filipa Rocha, Antonio José de Barros Neto, Hugo Simão, Hugo Nicolau, and Tiago Guerreiro. 2020. Exploring Accessible Programming with Educators and Visually Impaired Children. In *Proceedings of the Interaction Design and Children Conference (IDC '20)*. Association for Computing Machinery, New York, NY, USA, 148–160. https://doi.org/10.1145/3392063.3394437

[15] Jeannette Wing. 2008. Computational thinking and thinking about computing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 366 (11 2008), 3717–25. https://doi.org/10.1098/rsta.2008.0118

[16] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.

[17] Junnan Yu and Ricarose Roque. 2019. A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction* 21 (2019), 17 – 36. https://doi.org/10.1016/j.ijcci.2019.04.001