Design, development, and evaluation of child-robot interaction aimed at enhancing the development of computational thinking in preschool children

Ewelina Bakała

Postgraduate program PEDECIBA Informática
Instituto de Computación - Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
November of 2024

Design, development, and evaluation of child-robot interaction aimed at enhancing the development of computational thinking in preschool children

Ewelina Bakała

Doctorate's Thesis submitted to the Postgraduate Program PEDECIBA Informática, Instituto de Computación - Facultad de Ingeniería of the Universidad de la República, as part of the necessary requirements for obtaining the title of Doctor in PEDECIBA Informática.

Director:
Dr. Gonzalo Tejera

Academic director:
Dr. Héctor Cancela

Montevideo – Uruguay
November of 2024

Bakała, Ewelina

Design, development, and evaluation of child-robot interaction aimed at enhancing the development of computational thinking in preschool children / Ewelina Bakała. - Montevideo: Universidad de la República, Instituto de Computación - Facultad de Ingeniería, 2024.

IX, 181 p. 29, 7cm.

Director:

Gonzalo Tejera

Academic director:

Héctor Cancela

Tesis de Doctorado – Universidad de la República, Program PEDECIBA Informática, 2024.

Bibliography: p. 15 – 18.

1. Robótica Educativa, 2. Pensamiento Computacional, 3. Preescolares. I. Tejera, Gonzalo, . II. Universidad de la República, Postgraduate Program PEDECIBA Informática. III. Title

MEMBERS OF THE THESIS DEFENSE COURT

Ph.D. Prof. Taciana Falcão

Ph.D. Prof. Marcelle Mota

Ph.D. Prof. Pablo Ezzatti

Ph.D. Prof. Marcos Viera

Ph.D. Prof. Daniel Calegari

Montevideo – Uruguay
November of 2024

In order of appearance:
To Álvaro, Mateo, Pedro, and
Eva.

Acknowledgements

I am profoundly grateful to my advisor, Gonzalo Tejera, and to Juan Pablo Hourcade from the University of Iowa, who contributed to this research, for their support and guidance throughout this journey. Thank you for the countless meetings and the wealth of insightful feedback along the way.

I would like to extend my heartfelt thanks to the MINA Group for welcoming me as a member and supporting my research. I am also grateful to the educational institutions that allowed me to conduct the studies and to the teachers for their valuable time and collaboration.

Finally, to my husband, Alvarito—this work would not have been possible without your love, patience, and constant encouragement.

ABSTRACT

Computational thinking (CT) is a skill that enables individuals to formulate problems in a way that can be solved by computers. Interest in CT spans both academia, which annually produces a growing body of research on the topic, and educational settings that are integrating CT into curricula worldwide. While most educational initiatives focus on primary and secondary education, CT can also be effectively introduced at the preschool level, such as through the use of educational robots. This thesis compiles a series of studies that explore various aspects of teaching computational thinking at kindergarten level. We examine the current state of the art by investigating tools, activities, and CT evaluation methods, and then narrow our focus to the concept of conditionals—a programming concept that goes beyond simple sequencing and enables the construction of more complex algorithms. We evaluate current approaches to teaching conditionals, expand the programming language of Robotito, an educational robot developed in Uruguay, to include support for conditionals, and conduct a field study using the robot. Our findings not only highlight the potential of child-robot interaction in fostering early CT skills but also offer concrete developments, as well as design and evaluation methods, for future educational robotics initiatives aimed at preschoolers.

Keywords:
Educational Robotics, Computational Thinking, Preschoolers.

Contents

Chapter 1

Introduction

Computational thinking (CT) was first introduced by Jeannette Wing in 2006 as a skill that "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science." [29] Since then, the academic community has not reached a consensus on a single definition of CT or a unique list of abilities that it requires. A comprehensive review of CT definitions by Bocconi [23] showed that existing definitions fall into two main categories: those associating CT with generic problem solving concepts (e.g., abstraction, decomposition, generalization) and those linking CT to programming and computing (e.g., algorithmic thinking, data types, conditionals). Core concepts identified across both categories include abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization [9, 23]. Our work is guided by a broader definition, which describes computational thinking as a skill that enables individuals to approach and solve problems in a manner that can be implemented by a computer [28]. We found this definition more illustrative than a list of concepts, yet broad enough to encompass all core skills.

CT has gained increasing attention from policymakers due to the growing importance of digital literacy in the modern world [23]. As technology continues to permeate every aspect of our lives, from basic communication to complex scientific research, the ability to think computationally is becoming essential. Teaching CT at an early age, including at the preschool level, is crucial because it lays the groundwork for these essential skills early in a child's cognitive development. Early exposure could help children develop abilities that embrace sequencing, problem-solving or debugging [8]. By integrating

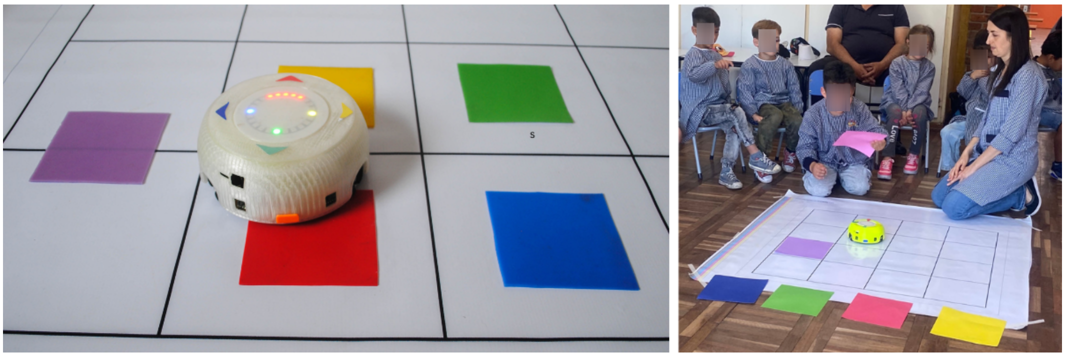


Figure 1.1: Left: Robotito and the color cards. The robot moves forward with red, left with blue, backward with green and right with yellow. Purple makes it spin. Right: An example of children solving a programming task.

CT into early childhood education, we can ensure that the next generation is not only proficient in using technology but also capable of innovating and creating new technological solutions.

Educational robotics (ER) provides hands-on, interactive way for young children to learn. Robots have proven to be effective tools for introducing abstract concepts at the preschool level, with empirical studies reporting their successful use in stimulating CT development in young children [1, 7, 11, 12, 15, 16, 18, 19, 22, 24, 26].

This favorable context motivated the development of Robotito, an educational robot aimed as a research platform and a tool to stimulate the development of CT in young children [27]. Designed and created at Universidad de la República in Uruguay, Robotito can be programmed by modifying its environment. It utilizes a color sensor located underneath the robot to detect color cards placed on the floor and moves accordingly based on the detected colors (see Figure 1.1). This setup allows children to program sequences of movements by arranging color patches on the floor. Robotito has proven to be a valuable tool for fostering CT development in young children [13].

Recognizing the importance of introducing CT concepts at an early age and the benefits of using educational robots for this purpose, this thesis explores various aspects of stimulating CT in preschoolers using robots. It presents findings from both literature reviews and empirical studies with commercial robots and Robotito.

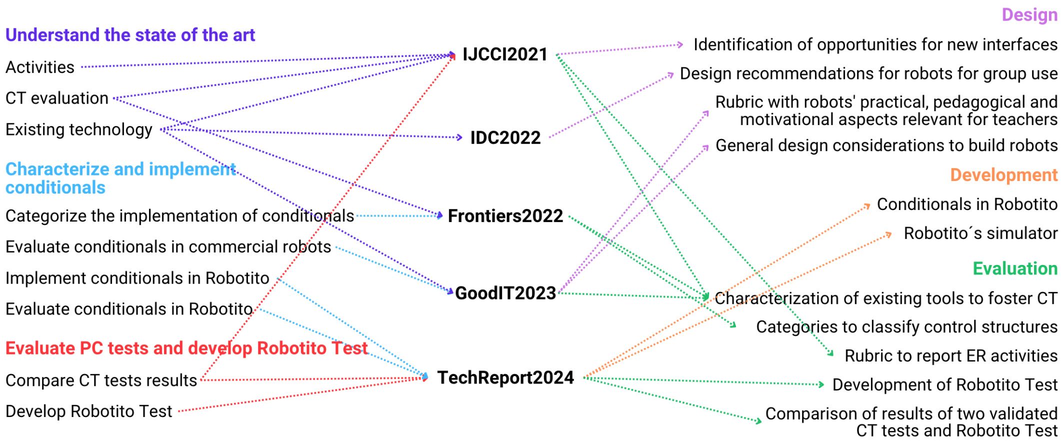


Figure 1.2: Objectives of this work on the left side, articles related to each objective in the middle, and contributions of the articles on the right.

1.1. Objectives

The common purpose of the works compiled in this thesis is to contribute to the design, development and evaluation of child-robot interaction, aimed at enhancing the development of computational thinking in preschool children. That main purpose is further decomposed into three specific objectives. These are outlined as follows:

Understand the state of the art: Provide a better understanding of experiences for prompting the development of CT in young children through systematic literature reviews and empirical studies of existing technologies. This includes analyzing activities, CT evaluation methods, and available tools.

Characterize and implement conditionals: Explore the implementation of conditionals in technology designed for preschoolers, integrate conditionals into Robotito, and assess whether young children can effectively use them.

Evaluate existing CT tests and develop Robotito Test: Evaluate existing CT tests and develop an evaluation to measure the impact of activities with Robotito.

1.2. Contribution

This thesis contributes to design, development and evaluation of child-robot interaction, aimed at enhancing the development of computational thinking in preschool children. Its contributions can be outlined as follows (see Figure 1.2 to visualize articles that led to the creation of each contribution):

Design

- Identification of opportunities for new interfaces.
- Design recommendations for building robots intended for group use.
- A rubric outlining practical, pedagogical, and motivational aspects of robots that are most important for classroom use, according to teachers.
- A compilation of general design considerations for developing robots for kindergarten.

Development

- Design and implementation of conditionals in Robotito.
- Design and implementation of Robotito's simulator.

Evaluation

- Characterization of existing tools to foster CT.
- Categories for classifying the implementation of control structures in tools designed for preschoolers.
- A rubric to report ER activities.
- Development of Robotito Test that evaluates childrens' knowledge about how to program Robotito.
- A comparison of results of two validated CT tests and Robotito Test.

1.3. Structure of Document

Besides the articles compiled in this thesis, this document lays out an introductory text to frame the work presented in them. Chapter 2 summarizes and places in context four published articles and a technical report composing this thesis. Finally, Chapter 3 provides global concluding remarks and discusses future work.

Chapter 2

Articles Supporting this Ph.D. Thesis

This chapter presents a collection of articles and a technical report that emerged from exploring robot-mediated stimulation of computational thinking in young children. The included works encompass systematic literature reviews and empirical studies involving both teachers and children (see Appendix 1 for details about institutions, teachers, and students who participated in the studies).

2.1. Articles Compiled in this Ph.D. Thesis

We used the acronyms of the journals and conferences from Figure 1.2 to label the subsections to make it easier to visualize the objectives and contributions of each article.

2.1.1. IJCCI2021

[6] Ewelina Bakala et al. «Preschool children, robots, and computational thinking: A systematic review». In: *International Journal of Child-Computer Interaction* 29 (2021), p. 100337

The first article is a systematic literature review of peer-reviewed publications that present educational robotics interventions aimed at promoting CT during early childhood (this article is included in Appendix 2). The purpose of this work is to provide an overview of the field by characterizing the robots used in the studies, the activities developed, and the evaluation methods applied to measure CT. Additionally, it analyzes the research contexts and the motivations behind conducting these studies.

This review identifies and characterizes robots with a measurable impact on CT development and highlights opportunities for new child-robot interfaces. It also points out reporting gaps and proposes a rubric to facilitate uniform and detailed reporting of ER activities. The study further acknowledges a lack of empirical research employing experimental or quasi-experimental designs and validated instruments to measure CT development, which motivated the work presented in the technical report that concludes this thesis.

2.1.2. IDC2021

[5] Ewelina Bakala et al. «Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten». In: *Proceedings of the 21st annual acm interaction design and children conference*. 2022, pp. 422–429

This article further explores robots developed for stimulating CT in young children, this time focusing on their application in classroom settings (this article is included in Appendix 3). This study presents findings from evaluations involving children and three robots, each with a distinct user interface, during group activities.

The research identifies key design factors affecting social use of these robots and provides design recommendations for designing programmable robots for classroom environments.

2.1.3. Frontiers2022

[4] Ewelina Bakala et al. «A Systematic Review of Technologies to Teach Control Structures in Preschool Education». In: *Frontiers in Psychology* 13 (2022), p. 911057

This systematic review initiates our exploration of control structures, and specially conditionals, in technology for young children (this article is included in Appendix 4). We consider conditionals an essential concept that enables children to progress beyond simple sequencing and develop more complex algorithms. They are critical component of many CT frameworks and definitions [10, 14, 25] and validated CT tests [21, 30].

The article is part of the Research Topic "Stem, Steam, Computational Thinking and Coding: Evidence-based Research and Practice in Children's Development" and presents the current state of the art of teaching control structures to kindergarten level children using electronic tools.

This work identifies 110 tools designed to teach CT to young children and analyzes the characteristics of those appropriate for our target age group. Additionally, it presents empirical evidence on the effectiveness of these tools in teaching control structures. The study also proposes categories to assess the tools' ability to express control structures and provides examples for each category.

The research highlights a scarcity of studies that assess the use or understanding of control structures. It also identifies a knowledge gap regarding how children develop early notions of control structures and which tools are most effective in introducing these concepts.

2.1.4. GoodIT2023

[3] Ewelina Bakala et al. «"It will surely fall": Exploring Teachers' Perspectives on Commercial Robots for Preschoolers». In: *Proceedings of the 2023 ACM Conference on Information Technology for Social Good.* 2023, pp. 477–486

To further understand the available tools and their suitability for classroom use, this article focuses on four commercial robots that support working with control structures (this article is included in Appendix 5). It explores teachers' perspectives on the advantages, challenges, and opportunities of implementing these tools in a preschool classroom.

The study provides a detailed evaluation of the robots and contributes with a rubric with practical, pedagogical and motivational aspects that should be taken into account while designing and evaluating robots. Additionally, it identifies general design considerations for developing robotic environments for kindergarten classrooms.

2.1.5. TechReport2024

[2] Ewelina Bakala, Gonzalo Tejera, and Juan Pablo Hourcade. *An Iterative Design and Empirical Evaluation of Conditionals for Robotito*. Tech. rep. Udelar. FI., 2024. URL: https://hdl.handle.net/20.500.12008/45833

The final work included in this thesis is a technical report describing the iterative design process conducted to develop conditionals for Robotito and the results of an ER intervention with Robotito (this article is included in Appendix 6). It responds to the lack of empirical studies with quasi-experimental design involving validated tools for CT evaluation reported in [6] and a scarcity of studies that asses the understanding of control structures pointed out in [4].

It outlines the evaluation process of various conditionals prototypes, culminating in the implementation of the most suitable option in the physical robot. Additionally, it introduces Robotito's simulator, which proved to be not only valuable in the evaluation process but also beneficial during classroom activities. The study also presents the Robotito Test, developed to asses children's knowledge about Robotito, as well as a unique comparison of the results of validated CT tests.

2.2. Statement of Authorship

The author of this thesis is the main author of the works presented in this chapter. A detailed description of the author's contribution to each piece can be found in the corresponding appendices.

None of the works in this compendium were included in other compendium-thesis documents, nor will they be included in such type of thesis in the future (see Appendix 7 for authorship statements of the co-authors).

Chapter 3

Conclusions and Future Works

The primary goal of this thesis was to contribute to the design, development, and evaluation of child-robot interaction aimed at stimulating the development of CT in preschoolers. The contributions of this work are grounded in literature reviews and empirical studies involving both teachers and children.

3.1. Concluding remarks

3.1.1. Design

Designing robotic platforms for classroom use remains an underexplored area that requires further research. While numerous robots have been developed to stimulate CT in young children (see [4, 6]), most existing tools appear to be designed for individual, at-home use rather than school settings. Our field study [5] identified problematic issues that raised when using commercial robots in group settings. Additionally, as we remarked in [6], the interfaces of robots used in the empirical studies we reviewed often seemed unsuitable for young children, particularly because their button-based interfaces are cognitively demanding and require high working memory load.

In [6], we identified several unexplored interfaces that should be tested in empirical studies. Furthermore, we contributed to the future design of robots for preschoolers and classroom context by offering design recommendations driven from field studies with children [5] and focus groups with teachers [3]. Although the results of these studies are based on work with a limited number of teachers and children, we hope they still offer valuable guidelines for future research to build upon. Additionally, we provided designers with a list of relevant practical, pedagogical and motivational aspects that should be considered when designing robots for preschoolers [3].

As emphasized in [6], there is a need for low-cost, open-source robots to provide a flexible platform for both teaching and research. We hope that our insights will help in the design of robots suitable for early childhood education and classroom use.

3.1.2. Development

This thesis also contributes concrete developments. To enable Robotito to support more complex programming concepts, we designed three different prototypes of conditionals, evaluated them with teachers, and implemented the most appropriate option in the physical robot [2]. The classroom implementation of activities involving conditionals, along with the results of CT tests and the Robotito Test, allowed us to conclude that conditionals can be successfully introduced at the preschool level.

The Robotito simulator, originally developed as a tool to evaluate new ideas for the robot, proved to be not only a valuable evaluation instrument but also a useful support for in-class activities [2].

Our developments led to an extension of Robotito's curriculum in two ways: first, by incorporating a new CT concept—conditionals; second, by providing a tablet-based application that allows children to experience more instances of programming the robot.

3.1.3. Evaluation

This work contributes to a better understanding of existing tools for stimulating CT development. In [4], we provided a comprehensive overview of tools reported in the literature. We categorized them and analyzed in detail those relevant to our study—electronic tools targeting our age group that do not require reading skills and allow for the construction of an explicit program. Our work helped to identify how these tools incorporate control structures into code and their effectiveness in teaching control structures, measured by empirical studies. Our findings indicated that only one study provides evidence of kindergarten children mastering conditionals, highlighting evaluation opportunities focused on control structures that should be explored in future research.

We also analyzed empirical studies to characterize robots that have demonstrated a positive impact on CT development [6]. We examined and discussed robots' input and output interfaces and identified opportunities for further research.

[5] extended our understanding of existing robots through in-field observations, while [3] helped to further characterize them, taking into account the perspective of experienced teachers. Both studies helped to identify aspects relevant to classroom use based on group work.

To further contribute to the evaluation and comparison of existing tools, in [4] we proposed novel categories to classify control structures, and in [3] we introduced a rubric addressing practical, pedagogical, and motivational aspects of robots, relevant to teachers, which enables the assessment of robots' suitability for classroom environments.

Our evaluation addressed not only tools, but also activities and demonstrated many reporting gaps that we addressed by proposing a rubric to report and evaluate CT activities [6]. This rubric was used to report a field study detailed in [2].

Another evaluation instrument developed during this Ph.D is the Robotito Test. It was used to assess children's understanding of Robotito programming after a series of ER activities with the robot [2]. Although it provided valuable data on children's understanding of conditionals, we identified issues related to its paper-based nature that should be addressed in future research.

Finally, this work offers a unique comparison of the results of two validated CT tests. The comparison revealed surprising differences in test scores that warrant further investigation in future studies.

3.2. Open Research Lines and Future Work

Although the compiled articles discuss unexplored areas and concrete directions for future works, we would like to conclude this thesis with a broader reflection on the most remarkable future works related to the design, development, and evaluation of child-robot interaction.

Regarding design and development, there is a need to create tools that better support group work. Classroom activities often require shared use of robots, not only for financial reasons (such as cost) and logistical considerations (like storage and maintenance), but also for pedagogical reasons (such as fostering collaboration). Therefore, it is essential to design robots that facilitate group work. Some initial studies have focused on collaborative interfaces for preschoolers [17, 20], but these efforts should be expanded in future research. The recommendations for developing technology suitable for classroom settings that we presented in [3, 5] can serve as a staring point, but they should be aligned to specific contexts through participative, iterative, and interdisciplinary research.

Additionally, the diversity of children should be carefully considered when designing collaborative environments, ensuring they are engaging and accessible for everyone involved. While Robotito demonstrated positive performance when children collaborated in small groups [5], several accessibility aspects still need attention. For example, the robot's programming is based on color cards with no tactile cues, the color-direction relationship is represented only visually, the robot cannot be easily lifted with one hand, and sound cues are used for the musical mode rather than as multimodal feedback. Addressing these aspects is essential to making Robotito a more versatile tool for inclusive classrooms.

We also observed that classroom settings often involve minimal adult supervision of a working group, highlighting the need for robots that can support and guide activities autonomously. Among the 24 robots with tangible user interfaces identified in [4], only Qobo, has the capability to guide activities. In its "game mode," Qobo senses the cards beneath it and executes the corresponding actions, providing audio feedback that helps children correctly orient the robot and initiate programming on the "Start" card. Teachers positively evaluated the robot's ability to offer feedback on children's performance [3], suggesting that this feature should be considered valuable in future designs.

In terms of evaluating child-robot interaction, our systematic literature review [6] highlighted the need for more empirical studies with experimental designs that utilize validated CT tests to assess the impact of educational robotics activities. Although our recommendation was based on data from articles searched in 2020, as of 2024, only five studies (discussed in [2]) have used validated CT tests for preschoolers. Moreover, four of these studies were conducted by the test authors, indicating a need for the broader research community to adopt these valuable evaluation tools in their practice.

The surprising and contradictory results of two validated CT tests in our field study [2] suggest the need for more comparative studies to explain these discrepancies. The lack of correlation between the tests, even when assessing similar CT concepts, raises questions about what aspects or levels of difficulty of each concept each test measures and in what context each test is most appropriate. Future research should address these questions to clarify the particularities, strengths and limitations of each assessment tool.

References

[1] Charoula Angeli and Nicos Valanides. «Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy». In: *Computers in Human Behavior* 105 (2020), p. 105954.

[2] Ewelina Bakala, Gonzalo Tejera, and Juan Pablo Hourcade. *An Iterative Design and Empirical Evaluation of Conditionals for Robotito*. Tech. rep. Udelar. FI., 2024. URL: https://hdl.handle.net/20.500.12008/45833.

[3] Ewelina Bakala et al. «"It will surely fall": Exploring Teachers' Perspectives on Commercial Robots for Preschoolers». In: *Proceedings of the 2023 ACM Conference on Information Technology for Social Good*. 2023, pp. 477–486.

[4] Ewelina Bakala et al. «A Systematic Review of Technologies to Teach Control Structures in Preschool Education». In: *Frontiers in Psychology* 13 (2022), p. 911057.

[5] Ewelina Bakala et al. «Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten». In: *Proceedings of the 21st annual acm interaction design and children conference*. 2022, pp. 422–429.

[6] Ewelina Bakala et al. «Preschool children, robots, and computational thinking: A systematic review». In: *International Journal of Child-Computer Interaction* 29 (2021), p. 100337.

[7] Marina U Bers, Carina González-González, and Mª Belén Armas–Torres. «Coding as a playground: Promoting positive learning experiences in childhood classrooms». In: *Computers & Education* 138 (2019), pp. 130–145.

[8] Marina Umaschi Bers, Amanda Strawhacker, and Amanda Sullivan. «The state of the field of computational thinking in early childhood education». In: 274 (2022). DOI: https://doi.org/https://doi.org/10.1787/3354387a-en. URL: https://www.oecd-ilibrary.org/content/paper/3354387a-en.

[9] Stefania Bocconi et al. *Developing computational thinking in compulsory education-Implications for policy and practice*. Tech. rep. Joint Research Centre (Seville site), 2016.

[10] Karen Brennan and Mitchel Resnick. «New frameworks for studying and assessing the development of computational thinking». In: *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*. Vol. 1. 2012, p. 25.

[11] Garry Falloon. «Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling». In: *Computers & Education* 216 (2024), p. 105045.

[12] Kyriakoula Georgiou and Charoula Angeli. «Developing preschool children's computational thinking with educational robotics: The role of cognitive differences and scaffolding». In: *16th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA 2019*. 2019, pp. 101–108. ISBN: 9789898533937. DOI: 10.33965/celda2019_2019111013.

[13] Anaclara Gerosa et al. «Cognitive abilities and computational thinking at age 5: Evidence for associations to sequencing and symbolic number comparison». In: *Computers and Education Open* 2 (2021), p. 100043. ISSN: 2666-5573. DOI: https://doi.org/10.1016/j.caeo.2021.100043. URL: https://www.sciencedirect.com/science/article/pii/S2666557321000148.

[14] Shuchi Grover and Roy Pea. «Computational thinking in K-12: A review of the state of the field». In: *Educational researcher* 42.1 (2013), pp. 38–43.

[15] Kay Yong Khoo. «A Case Study on How Children Develop Computational Thinking Collaboratively with Robotics Toys». In: *International Journal of Educational Technology and Learning* 9.1 (2020), pp. 39–51.

[16] Tess Levinson and Marina Bers. «Robotics in universal prekindergarten classrooms». In: *International Journal of Technology and Design Education* (2024), pp. 1–22.

[17] Yanhong Li et al. «Designing a Tangible Interface to "Force" Children Collaboration». In: *Proceedings of the 21st Annual ACM Interaction Design and Children Conference*. 2022, pp. 576–582.

[18] Ana García Valcárcel Muñoz-Repiso and Yen Air Caballero-González. «Robotics to develop computational thinking in early Childhood Education». In: *Comunicar* 27.59 (2019), pp. 63–72. ISSN: 19883293. DOI: 10.3916/C59-2019-06.

[19] Ki Won Nam, Hye Jeong Kim, and Suyoun Lee. «Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners». In: *The Asia-Pacific Education Researcher* 28.5 (2019), pp. 387–397. ISSN: 2243-7908. DOI: 10.1007/s40299-019-00438-4. URL: https://doi.org/10.1007/s40299-019-00438-4.

[20] Kryscia Ramírez-Benavides, Gustavo López, and Luis A Guerrero. «Designing tools that allows children in the early childhood to program robots». In: *Learning and Collaboration Technologies. Technology in Education: 4th International Conference, LCT 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 4*. Springer. 2017, pp. 71–89.

[21] Emily Relkin and Marina Bers. «Techcheck-k: A measure of computational thinking for kindergarten children». In: *2021 IEEE global engineering education conference (EDUCON)*. IEEE. 2021, pp. 1696–1702.

[22] Evgenia Roussou and Maria Rangoussi. «On the use of robotics for the development of computational thinking in kindergarten: educational intervention and evaluation». In: *International Conference on Robotics and Education RiE 2017*. Springer. 2019, pp. 31–44.

[23] Bocconi S et al. *Reviewing Computational Thinking in Compulsory Education*. Scientific analysis or review KJ-06-22-069-EN-N (online). Luxembourg (Luxembourg), 2022. DOI: 10.2760/126955(online).

[24] Anika Saxena et al. «Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education». In: *The Asia-Pacific Education Researcher* 29.1 (2020), pp. 55–66.

[25] Valerie J Shute, Chen Sun, and Jodi Asbell-Clarke. «Demystifying computational thinking». In: *Educational research review* 22 (2017), pp. 142–158.

[26] Jihyun Sung, Ji Young Lee, and Hui Young Chun. «Short-term effects of a classroom-based STEAM program using robotic kits on children in South Korea». In: *International Journal of STEM Education* 10.1 (2023), p. 26.

[27] Gonzalo Tejera et al. «Robotito: programming robots from preschool to undergraduate school level». In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 296–301.

[28] Jeannette Wing. «Computational thinking's influence on research and education for all». In: *Italian Journal of Educational Technology* 25.2 (2017), pp. 7–14.

[29] Jeannette M Wing. *Computational thinking*. 2006.

[30] María Zapata-Cáceres, Estefanía Martín-Barroso, and Marcos Román-González. «Computational thinking test for beginners: Design and content validation». In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2020, pp. 1905–1914.

APPENDICES

Appendix 1

Information about institutions, teachers and students that participated in the empirical studies

Institution	Children	Article id
Jardín 238	15 children aged 5 to 6	IDC2021
Jardín 216	56 children aged 5 to 6	TechReport2024

Table 1.1: Educational institutions and children who participated in the empirical studies.

.

Teacher's id	Description	Reference Institution	Article id
Teacher 1	Teacher who teaches at both public and private institutions at the preschool and primary school levels	The Anglo School	GoodIT2023 (T1) TechReport2024 (P2)
Teacher 2	Computing teacher who works with preschoolers and early primary school students	The Anglo School	GoodIT2023 (T2) TechReport2024 (P1)
Teacher 3	Preschool teacher from a public institution	Jardín 345	TechReport2024 (P3)

Table 1.2: Teachers who participated in the empirical studies. The identifiers referenced in the articles are provided in brackets for clarity.

.

Appendix 2

Preschool children, robots, and computational thinking: A systematic review

Ewelina Bakala et al. «Preschool children, robots, and computational thinking: A systematic review». In: *International Journal of Child-Computer Interaction* 29 (2021), p. 100337

Author's contribution The author conceptualized the research idea, designed the methodology, and developed the data collection strategies. She was one of the two reviewers who conducted the study selection process, as well as data extraction and systematization. She also led and actively participated in the writing process.

International Journal of Child-Computer Interaction 29 (2021) 100337

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

Preschool children, robots, and computational thinking: A systematic review

Ewelina Bakala [a,*], Anaclara Gerosa [b], Juan Pablo Hourcade [c], Gonzalo Tejera [a]

[a] INCO, Universidad de la República, Uruguay
[b] CICEA, Universidad de la República, Uruguay
[c] Department of Computer Science, The University of Iowa, USA

ARTICLE INFO

Article history:
Received 14 November 2020
Received in revised form 21 May 2021
Accepted 24 May 2021
Available online 31 May 2021

Keywords:
Educational robotics
Computational thinking
Preschool

ABSTRACT

We conducted a systematic review of empirical studies aimed at exploring robot-mediated activities to promote the development of computational thinking in preschoolers. In this study, we investigated the robots used, proposed activities, and evaluation processes. We also analyzed research contexts and the stated motivations to conduct the studies. Our review identified characteristics of the robots, such as input and output interfaces, cost, and availability. We also categorized activities considering context, modality of work, type of activities, duration, adults' role, scaffolding, unplugged activities, explicit debugging, communication and sharing instances, and teaching knowledge from other domains. We analyzed the computational thinking evaluation process looking at types of assessments, asset concepts, and research design. This paper presents a comprehensive overview of existing research, identifies existing gaps, and provides recommendations for future studies.

Contents

* Corresponding author.
E-mail address: ebakala@fing.edu.uy (E. Bakala).

https://doi.org/10.1016/j.ijcci.2021.100337

1. Introduction

In recent decades, technological developments related to computing, informatics, and digitization have generated radical changes in our lives. Most of us interact with digital technology daily and rely upon it to perform most of our tasks.

The term "computational thinking" (CT) has reemerged in recent literature as a way to describe several skills required to formulate, model, and solve problems using strategies and ideas from computer science (Barr, Harrison, & Conery, 2011; Cuny, Snyder, & Wing, 2010). Despite it still being a term under construction, many authors include algorithmic thinking, abstraction, decomposition, sequencing, generalization, and debugging as part of CT skills (Grover & Pea, 2013). While CT has sometimes been equated to programming, several authors emphasize CT as a broader cognitive skill set rather than the specific action of coding (Shute, Sun, & Asbell-Clarke, 2017), a trend started by Wing in her influential 2006 article (Wing, 2006) in which she proposed that "Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability".

CT has sparked educators, practitioners, and policy-makers' interest and has been included in curricula worldwide (Bocconi et al., 2016; So, Jong, & Liu, 2020; Uscanga, Bottamedi, & Brizuela, 2019). Despite much interest, many questions regarding the integration of CT in education and the best practices for teaching and learning remain unanswered. For example, stimulating CT development at an early age continues to be an academic challenge whose approach requires research, interdisciplinary work, and innovation.

During early childhood, the development of abilities such as self-regulation, working memory, and inhibitory control increases exponentially, thus establishing this stage as a window of opportunities for interventions aimed at promoting child development (Tsujimoto, 2008). Several authors have pointed out early childhood education programs (Campbell et al., 2014; Doyle, Harmon, Heckman, & Tremblay, 2009; Heckman & Carneiro, 2003) yield high return rates for government and society through a long-term impact on participants' health and overall quality of life. Recent work (Bers, 2018, 2020; Botički, Kovačević, Pivalica, & Seow, 2018) indicates promoting CT at an early age enhances children's analytical capacities and introduces them to new mental tools that are useful for collaborative problem solving and expression. Moreover, Bers considers the early learning of CT as part of the positive technological development framework, which is grounded on the notion of promoting the use of technology to support positive interpersonal behaviors in children growing up in the digital age (Bers, 2010).

Our research is also motivated by Uruguay's particular context: since 2007 Plan Ceibal (Plan Ceibal) (started as a Uruguayan adaptation of the One Laptop per Child project (One Laptop Per Child, n.d.)) has promoted children's access to Information and Communication Technologies (ICTs) as a state policy. Its objective is to support educational goals through the use of ICTs while promoting equity and inclusion. Plan Ceibal has provided every public primary school student with a laptop and set up high-speed Internet access in every public primary school to achieve this objective. Recently, it began to include CT courses for primary school children. To support this initiative and explore the possibility of teaching CT using robots, our research team developed a robot called Robotito that defines its behavior according to the physical disposition of the elements in its environment (Tejera et al., 2019). Robotito was used in a controlled intervention with primary school children using an experimental design and preliminary results were promising for children who presented a high level of engagement with these tasks (Gerosa, Koleszar, Gómez-Sena, Tejera, & Carboni, 2019). This review is an initial step in the adaptation of Robotito's design and capabilities to a preschool context to provide teachers with a developmentally appropriate tool to promote the development of CT among the youngest students.

1.1. Related works

Previous evidence (Bers, Flannery, Kazakoff, & Sullivan, 2014; Toh, Causo, Tzuo, Chen, & Yeo, 2016) suggests CT could be integrated in developmentally appropriate ways early on through robots, which provide an attractive and motivating way for children to access technology and promote CT through playful learning. There is also evidence that having tangible, physical outputs in CT activities could be advantageous. For example, Almjally, Howland, and Good (2020) found that young children who use body gestures during programming activities learned more programming concepts. The authors reported that the children frequently used pointing gestures to simulate the robot's actions. It is possible that movement in a robot could make it easier for children to perform such gestures. Physical outputs can also lead to greater class engagement (Zhu, Ma, Wong, & Huen, 2016) and may thus be better suited for school-based activities than CT activities with visual outputs. However, there is no systematization of the information about research that used robotic platforms to stimulate preschoolers' CT development.

While systematic reviews have been conducted to explore existing computational kits (Hamilton, Clarke-Midura, Shumway, & Lee, 2020; Yu & Roque, 2019), the use of robots in education (Anwar, Bascou, Menekse, & Kardgar, 2019; Benitti, 2012; Jung & Won, 2018a; Kubilinskiene, Zilinskiene, Dagiene, & Sinkevičius, 2017) or learning CT through robotics (Ioannou & Makridou, 2018; Shute et al., 2017), there have been fewer works (Isnaini & Budiyanto, 2018; Umam, Budiyanto, & Rahmawati, 2019) which encompass both of these subjects simultaneously while targeting specifically the early childhood period. Additionally, existing works do not present an in-depth analysis of the current state of the art in this field. Isnaini and Budiyanto's (Isnaini & Budiyanto, 2018) results are based on the analysis of a wide age range of participants and do not provide specific conclusions or recommendations for early childhood education. Umam et al. (2019) focus on evaluating commercially available robotics devices and do not provide information about the activities they were involved in or metrics used to evaluate the robots' impact on CT development. We aim to provide a systematic review of peer-reviewed publications that present educational robotics (ER) interventions and experiences that promote CT during early childhood. Our work is motivated by the following research questions:

- RQ1: What kind of robots were used in the studies, and how can they be classified?
- RQ2: What are the characteristics of the activities that aim to stimulate the development of computational thinking?
- RQ3: How was computational thinking evaluated?
- RQ4: Which individuals and countries are most active and influential in research on computational thinking development for preschool children mediated by robots, and what have been their motivations for conducting research in this area?

2. Methodology

We carried out a systematic literature review (SLR) because this method makes the review procedures as objective, analytical, and repeatable as possible (Kitchenham, Budgen, & Brereton, 2015). We conducted a particular type of systematic review called mapping study (Kitchenham et al., 2010) (or mapping review according to Grand and Booth's classification (Grant & Booth, 2009)). Mapping studies are used to survey the available knowledge about a specific topic, contrary to conventional SLRs that try to answer a specific research question (Kitchenham et al., 2010). They use the same methods for searching and data extraction as conventional SLRs, but the quality assessment is more relaxed and research questions more coarse-grained as they tend to explore available information about the topic. Four reviewers participated in the review process. Two reviewers selected the studies and extracted the data. The other two were supervisors who validated papers considered marginal or about which the first two reviewers were uncertain and provided data extraction and analysis guidelines.

2.1. Search strategy

We used an automated search (Kitchenham et al., 2015) to find all articles related to the topic and carried out a manual review of the articles obtained. The first author defined the search criteria and conducted the search process.

To build the search term, we defined three keywords relevant for the review: robot, computational thinking, and preschool education, and identified their synonyms. The search terms identified were: robot*, computational thinking, preschool*, young children, early age*, kindergarten, lower education, childhood, early years, elementary education, young learner*. We reviewed terms used in titles and abstracts of the papers obtained in previous exploratory searches to ensure that they were included in our list.

We restricted our search to journals and conference papers. We did not specify the time period for the search because computational thinking is quite a new term, coined by Jeannette Wing in 2006 (Wing, 2006), and the time restriction was not necessary. We used Scopus, IEEE XPLORE, ScienceDirect, SpringerLink, and ACM search engines. The search term, search parameters, the date of the search, and the number of items found are presented in Table C.6 in Appendix C.

A set of three predefined articles (Bers et al., 2014; González & Muñoz Repiso, 2018; Isnaini & Budiyanto, 2018) was used to validate the search result's completeness. The predefined articles were obtained through an informal manual search, including the articles by well-known researchers in the area, and using personal knowledge. We achieved a recall of 100% (3 of 3 predefined articles were found).

2.2. Study selection

We defined the following inclusion criteria for the studies' selection:

- Publications including an empirical study
- Publications involving teaching and learning skills related to computational thinking[1] using robots.
- Publications reporting studies that include children between three and five years old, including six years old, if attending pre-primary school educational level.
- Articles published in journals and conferences.

Exclusion criteria were:

- Irrelevant nature of articles.
- Studies that focus on children outside of the target age range.
- Studies that do not use robots.
- Studies with no computational thinking evaluation.[2]
- Publications written in a language other than English.
- Papers that report experiences with children with neurodevelopmental disorders.
- Additional reports on the same study (only the most comprehensive was included).

Two reviewers applied the selection criteria independently to the articles and, based on their title and abstract, classified them as "relevant" or "irrelevant". The results were compared, and the articles that were classified differently among reviewers were discussed. If the consensus was not met or both reviewers agreed that the article needed a detailed analysis, a full-text revision of the article was conducted to decide the inclusion. After full-text revision, both reviewers analyzed the questionable items again. If they could not decide if the article should be included, the other two reviewers were asked to review the item and express their opinion.

[1] There is no unique definition of CT (Grover & Pea, 2013; Ioannou & Makridou, 2018; Shute et al., 2017), so we decided not to restrict the inclusion criteria to concepts or skills related to one particular definition and accepted all the definitions presented by the authors.

[2] The form of the CT evaluation was not restricted and we consider as valid all evaluation forms applied by the authors. For more details of the types of assessments used in the reviewed studies see Section 3.4.1.

2.3. Data extraction

For data extraction, we used a spreadsheet that included the publication details (Authors, Title, Year, Source title, DOI, and Abstract) for each paper and the information needed to answer the research questions. Our approach was a thematic synthesis, where data is tabulated in a way that is consistent with the research questions.

During data extraction, based on the concrete information provided in the publications, important trends and categories emerged. This fact led to the extraction form's adjustment to facilitate data collection (see Appendix A for the fields used in the final extraction spreadsheet). Independent data extraction by two reviewers was followed by reconciliation through discussion or moderation.

In some cases, the data provided in the article was insufficient to respond to the research questions. In those cases, additional procedures were implemented. These included: emails to authors or searching for additional information on the web (e.g., searching for robot characteristics, such as input interface and feedback given to the user).

3. Findings and implications

In this section, we present both results related to the selection process and research questions. As we examine each research question, we provide recommendations based on our findings.

3.1. Selection process result

We obtained 256 articles (62 from SCOPUS, nine from IEEE XPLORE, seven from ScienceDirect, 172 from SpringerLink, and six from ACM) through automated search. In the first step, repeated articles were excluded. We analyzed a total of 221 unique articles during the selection process. After the selection process (see Fig. 1), we identified 15 studies as relevant for this review (see Table C.7 in Appendix C for the publications included). Cohen's kappa coefficient (Cohen, 1960) was used to measure the level of inter-rater agreement in the screening step, the first selection step in which only titles and abstracts were analyzed. We achieved a coefficient of 0.96, indicating a high level of agreement between reviewers (Kitchenham et al., 2015). The final step of the selection was an iterative process based on full-text analysis, additional information provided by authors of some of the articles, and group discussion between the authors, hence the inter-rater reliability was not assessed.

3.2. RQ1: What kind of robots were used in the studies, and how can they be classified?

Table 1 presents an overview of the robots used in the studies. The most commonly used robot was the Bee-Bot. Next in popularity, used in four studies, were the following LEGO kits: LEGO WeDo, Mindstorm RIS, and Mindstorm NXT. In the case of one publication, the LEGO kit name was not specified. Two studies used KIBO, and one study used KIBO's predecessor KIWI. Two studies used Colby mouse. TurtleBot and Ozobot Bit were each used in one study.

3.2.1. Child–robot input interface

We identified three groups of user interfaces:

- **Physical buttons on the top of the robot.** Interfaces where each button is associated with a single movement of the robot, such as move forward, turn left, and so forth. They only allow for the definition and execution of a sequence of movements. They were used with the Colby mouse, Bee-Bot, and LEGO.

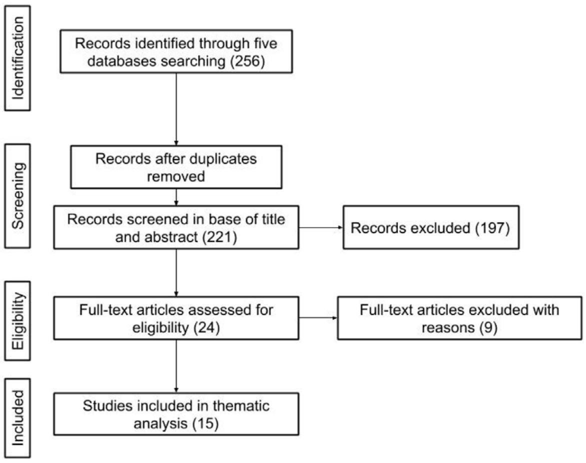


Fig. 1. Steps of the selection process.. *Source:* Adopted and adapted from Moher, Liberati, Tetzlaff, and Altman (2009).

- **Tangibles.** Interfaces where commands are codified by colored materials (Ozobot and TurtleBot) or blocks scanned by the robot (KIBO and KIWI). Tangible material can be associated with a robot's action (Ozobot, TurtleBot, KIBO, KIWI) or with the program's flow (KIBO and KIWI; for example, a block represents the beginning of a loop).
- **Hybrid.** Tangible and graphical interfaces that allow children to create programs to control their robots using tangible wooden blocks and/or graphical on-screen icons.

Recommendation for future research: We observed that in half of the studies, the children program the robots using physical buttons. Physical buttons do not generate the structured code needed for debugging and iterations over the code. This kind of user interface makes programming cognitively demanding due to the high load on working memory, in which the child has to remember the entire sequence of commands that should be introduced. This type of interface also limits the activities to sequencing tasks, making it difficult to work on other skills related to CT. Some of these characteristics might seem counterintuitive for this age range, so further studies should be conducted to define this user interface's appropriateness in early childhood education.

All of the described user interfaces present a very low level of embodiment during programming. Motor skills are fundamental in early childhood, as they contribute to children's autonomy and exploration capacities. Motor development, particularly fine motor skills, is closely related to cognitive development (Ahnert, Schneider, & Bös, 2010; Oberer, Gashaj, & Roebers, 2017; Rhemtulla & Tucker-Drob, 2011; van der Fels et al., 2015). Furthermore, evidence suggests learning is facilitated through embodied experiences (Lozada & Carro, 2016; Macedonia, Müller, & Friederici, 2011), and relations between early exploratory motor competence and later academic achievement have been reported (Bornstein, Hahn, & Suwalsky, 2013). It has been proposed that higher levels of embodiment within a group task might promote recall during learning (Sullivan, 2018), and young children's use of body gestures during programming is correlated with better CT learning outcomes (Almjally et al., 2020). Thus, it was surprising that almost all robots, except for Ozobot, which can be programmed by drawing colorful lines on the floor, are programmed in a reduced physical space. Although Ozobot offers the possibility to engage in more spatially distributed programming, the research

Table 1
Summary of the applied robots.

Name	Online information	Cost (USD)[a]	Input interface	Available actions[b]	Reference
Bee-Bot	https://www.terrapinlogo.com/bee-bot-family.html	90	Physical buttons	sound, light, movement	Angeli and Valanides (2020), Georgiou and Angeli (2019), González and Muñoz Repiso (2018), Muñoz-Repiso and Caballero-González (2019), Saxena, Lo, Hew, and Wong (2020)
Colby mouse	https://www.learningresources.com/stem-robot-mouse	30	Physical buttons	sound, light, movement	Khoo (2020), Roussou and Rangoussi (2019)
Ozobot Bit	https://files.ozobot.com/stem-education/educator-botcamp.pdf	116	Tangibles (color)	light, movement	Khoo (2020)
TurtleBot	http://robomation.net/?page_id=1576	99	Tangibles	sound, light, movement	Nam, Kim, and Lee (2019)
KIBO	https://kinderlabrobotics.com/kibo/	220 – 590	Tangibles	sound, light, movement	Bers, González-González, and Armas-Torres (2019), Pugnali, Sullivan, and Bers (2017)
LEGO	No information about the LEGO kit	–	Hybrid	light, movement	Bers et al. (2014)
KIWI	https://ase.tufts.edu/devtech/readyforrobotics/research.html	–	Tangibles	sound, light, movement	Sullivan and Bers (2016)
LEGO wedo	https://education.lego.com/en-gb/product/wedo-2	215	Hybrid	light, movement	Kazakoff, Sullivan, and Bers (2013)
LEGO Mindstorms RIS	https://www.lego.com/cdn/product-assets/product.bi.core.pdf/4129439.pdf	195	Hybrid	movement	Sullivan and Bers (2013)
LEGO Mindstorms NXT	https://www.lego.com/cdn/product-assets/product.bi.core.pdf/4589647.pdf	470	Physical buttons	movement	Cho and Lee (2017)

[a]The cost was defined based on Amazon prices if there was no information on the official website.
[b]The actions were determined based on available online information.

that used Ozobot only included activities where the robot was programmed in a worksheet that does not offer the possibility of a high level of embodiment. We consider that more activities and robot programming interfaces that favor bodily activity, displacement, and spatially distributed programming should be developed and evaluated with preschoolers. Initial steps have already been taken in this direction (Bakała et al., 2019; Gerosa et al., 2019; Tejera et al., 2019).

There are many user interfaces not yet evaluated when programming robots in a preschool context, for example, programming by demonstration (Frei, Su, Mikhak, & Ishii, 2000; Raffle, Parkes, & Ishii, 2004), gesture-based user interfaces (Merkouris & Chorianopoulos, 2019; Pons & Jaen, 2019), and voice user interfaces (Poncela & Gallardo-Estrella, 2015). Currently available computational kits use mostly tangibles or graphical interfaces to support programming and alternatives like gestures and body movements are not provided (Yu & Roque, 2019). We do not know if they are considered not appropriate to the context of interest because they do not generate structured code that can be analyzed and improved, or there was no research made to analyze their potential. There should be more exploration in this area to evaluate this aspect, especially for children aged three for whom only two types of user interfaces (physical buttons (González & Muñoz Repiso, 2018; Muñoz-Repiso & Caballero-González, 2019; Saxena et al., 2020) and tangibles (Bers et al., 2019)) were reported (see Table C.7 in Appendix C to consult the age range of the children that participated in each study). There is evidence that voice user interfaces can promote peer interactions at the age of three to four (Superti Pantoja, Diederich, Crawford, & Hourcade, 2019) and bringing them into activities aimed at promoting CT could help combine social interaction with cognitive stimulation. As interface design shapes "what users are able to do and how they are able to do it" (Weintrop & Wilensky, 2018), the exploration of new forms of programming seems to be an important task that could help to provide preschoolers with appropriate tools for their initial steps in CT education.

3.2.2. Available robot actions

We identified three types of feedback provided by the robots: sound, light, and movement. Available robot actions were determined based on the publications' information and additional online research in official websites and user manuals.

The feedback explicitly reported in all the studies is the displacement of the robot. Only one study (Sullivan & Bers, 2016) reports sound reproduction (robot "sings") as a part of a programming challenge. Three studies (Bers et al., 2014, 2019; Sullivan & Bers, 2016) stated that the children learned to turn the robot's light on during the activities.

Recommendation for future research: In the reviewed studies, programming robots is mainly restricted to defining their movements. In many cases, children are expected to understand concepts related to directions, such as "turn left" or "turn right". Saxena et al. (2020) report that children aged three to four had difficulties using directional language when programming robots. Silvis, Lee, Clarke-Midura, Shumway, and Kozlowski (2020) observed that arrows that usually code the robots' act of turning were misinterpreted by the kindergartners and turning without advancing (robot turns in place and does not advance to adjacent cell) was counterintuitive for them even after multiple experiences coding robots. Yu and Roque (2019) pointed out that most programming commands used in computational kits involve motion and that it would be interesting to explore computational concepts through, for example, visual, auditory, or tactile programmable objects. We consider that programming other forms of robots' expressivity (e.g., light and sound) could be not only interesting but also beneficial, especially for the youngest programmers struggling with distinguishing left versus right and understanding the robots' way of turning.

3.2.3. Cost and availability

All the robots used were commercial robots. Their cost lies between 30 and 470 USD. The least expensive ones (ColbyMouse, Bee-bot, and TurtleBot) offer only a predefined, fixed way of programming, meaning that they come with a set of rules that cannot be changed. The only robot's action that the user can control is the movement's direction, and only predefined sequences of movements can be executed. They do not allow users to control the execution flow with control structures or loops. Although these robots provide light and sound feedback, this feedback cannot be controlled by the user. More expensive robots allow more sophisticated actions and control over output parameters, such as sound and light. Although they are less flexible and more cognitively demanding when programming, the least expensive robots were used in half of the studies. This fact raises the question of whether the robots were selected based on their appropriateness or their cost.

Recommendation for scientific community: To our knowledge, there is no low-cost, open-source, and open hardware platform that could be easily used by early childhood teachers or researchers. We consider that more effort should be employed to build a scientific community that could design, deliver, and maintain stable robotic platforms appropriate for early childhood education and research.

3.3. RQ2: What are the characteristics of the activities that aim to stimulate the development of computational thinking?

An overview of the data extracted to respond to RQ2 is presented in Table 2. Observations and recommendations that emerged during the analysis of the activities were grouped into the following themes: context, modality of work, type of activities, duration, adults' role, scaffolding, unplugged CT activities, explicit debugging, communication and sharing, and teaching contents from other domains.

3.3.1. Context

Through our analysis of the activities proposed for the ER interventions, we found that most studies were conducted in formal school settings. Only two studies took place in an informal setting: a summer program (Pugnali et al., 2017) and a summer enrichment course (Saxena et al., 2020).

Recommendation for scientific community: Formal contexts seem to be the most frequent scenario for activities focused on CT stimulation, not only in studies with robot-mediated activities (Ioannou & Makridou, 2018), but also in broader contexts (Hsu, Chang, & Hung, 2018). It is essential to acknowledge that it would have been impossible to get to the current state of research without the support given by schools and related government entities to implement research activities in formal contexts. We should contribute to a constant dialog between policy-makers and scientists to communicate the importance of the given support and provide evidence of the benefits of the collaboration for children and society.

3.3.2. Modality of work

Most studies were carried out as group-based robotics activities, with just a few reporting only individual activities. Two studies did not include information about whether the activities were proposed to children in groups or individually.

Recommendation for reporting and future research: We consider that the information about group configuration during the activities is a fundamental datum that should be reported in all the studies. In some cases, the authors had to infer whether children participated in the activities individually or in groups based on pictures or scarce information like mentions of "large

Table 2
Summary of the activities.

Authors	Context	Modality of work	Type	Includes free-play	Total duration	Scaffolding			Unplugged activities	Explicit debugging	Sharing moments
						Narrative	Objects	Embodied activities			
Angeli and Valanides (2020)	Formal	Individual	Mixed	Yes	1h 20 min	Yes	Yes				
Roussou and Rangoussi (2019)	Formal	Group-based	Mixed	Yes	–	Yes	Yes	Yes	Yes		
Khoo (2020)	Formal	Mixed	Goal-oriented		–	Yes	Yes				
Nam et al. (2019)	Formal	Group-based	Goal-oriented		–		Yes		Yes		
Bers et al. (2019)	Formal	Mixed	Mixed	Yes	12 h	Yes			Yes		Yes
Georgiou and Angeli (2019)	Formal	Individual	Goal-oriented		Variable		Yes				
Muñoz-Repiso and Caballero-González (2019)	Formal	Group-based	Goal-oriented		1 h	Yes	Yes			Yes	
González and Muñoz Repiso (2018)	Formal	Group-based	Goal-oriented		28 h	Yes					
Pugnali et al. (2017)	Informal	Individual	Mixed		8 h	Yes		Yes			Yes
Bers et al. (2014)	Formal	Mixed	Mixed	Yes	15 h				Yes	Yes	Yes
Saxena et al. (2020)	Informal	Group-based	Goal-oriented		20 h	Yes	Yes		Yes		
Sullivan and Bers (2016)	Formal	Group-based	Mixed		10 h						
Kazakoff et al. (2013)	Formal	–	Goal-oriented		8 h						
Sullivan and Bers (2013)	Formal	Group-based	Mixed		16 h	Yes					
Cho and Lee (2017)	Formal	–	Goal-oriented		20 h	Yes		Yes	Yes		

group and individual activities" (Pugnali et al., 2017) without specifying which type of activities used the robot. To decide if the activities can be successfully adapted into real classroom contexts or to correctly replicate studies, the information about group size is essential.

Another interesting aspect related to group size that needs further research is the definition of the ideal number of children per group for a given activity. The identification of this number is not a trivial task as it requires a balance between viability in the school context and the participation of all group members.

3.3.3. Type of activity (goal-oriented, open-ended, free play)

All the activities were guided by teachers or researchers who defined their structure and goal (labeled in Table 2 as "goal-oriented"). Some studies also included open-ended activities and/or free play ("mixed").

No studies evaluated if the presentation of the educational content impacts children's learning process. We observed frequent usage of narrative, presence of free-play, open-ended and problem-based activities, and art-related projects, but it is not clear how to present the activities to ensure that they are suitable and engaging for children's context.

Recommendation for future research: The studies presented in this review are empirical studies that attempt to measure the impact of robot-mediated activities in CT development. Given the nature of the studies, it is expected that most of them include goal-oriented activities designed to prompt CT. Additionally, CT seems to be taught mainly using problem-based and project-based learning strategies (Hsu et al., 2018). However, if our goal is to define activities appropriate for use in a school context during a sustained period of time, aspects such as motivation, engagement, and empowerment should also be considered. Providing students with opportunities to express themselves in a creative way by integrating diverse disciplines, such as, science, art, engineering, and design is considered an important challenge (Resnick & Rusk, 2020; Yu & Roque, 2019). Many computational toys and kits support storytelling and some of them can be decorated with arts and crafts (Yu & Roque, 2019). Taking into account these existing possibilities, we consider that more work should be conducted to determine how activities' presentation and adjustment to children's gender, abilities, socioeconomic, cultural context, and even generational differences can influence learning and define guidelines that will help to present the educational content. Previous work has called for more research on the way these factors could impact learning through educational robotics activities (Jung & Won, 2018b). Some initial efforts have been made to clarify these points at the Kindergarten level (Lee et al., 2020; Nicholson, 2019) and valuable insights can be extracted from previous experiences with older children, which could be adapted and validated for this particular age group (Komis, Romero, & Misirli, 2016; McLean & Harlow, 2017).

3.3.4. Duration

All of the reported activities took less than 30 h of total time of intervention. The total time ranged between 1 h and 28 h. Many activities took less than 10 h, and two studies reported results after less than 2 h of intervention.

It was not always possible to determine how the activities were distributed in time, for example, the duration of each session, how many sessions per week were conducted, how they were spaced in time (i.e., how much time passed between one session and the following).

Recommendation for reporting and future research: We observed the same problem with reporting pointed out by Ioannou and Makridou (2018)- the lack of information about the duration of activities in relation to learning goals, which makes follow-up studies and replication impossible. The variability in the activities' total duration indicates that there is no agreement on the amount of time children need to participate in activities in order to have a significant learning outcome. It would be useful to know the minimum amount of total time, the ideal duration of individual sessions, and their frequency for an intervention required to obtain specific CT results taking into account the developmental characteristics of this specific age range. We consider clarifying these aspects as an essential point for further research as it would have direct implications on any possible implementation in formal educational settings. Although some studies indicate that the average attention span for preschoolers is 22 min (DiCarlo, Pierce, Baumgartner, Harris, & Ota, 2012), this number was estimated only for the whole group dynamic and was not measured for robot-mediated activities.

We also observed that in both studies with less than 2 h of intervention, the children worked individually with the robot. It is possible that working individually with children takes more time and effort and, therefore, makes the interventions shorter and not so frequent. Another research question is to learn about differences in the acquisition of CT when working individually or in groups. This insight could help to optimize the use of time and human resources in future research.

3.3.5. Adults' role

We were interested in learning about adults' level of participation in the activities. It is crucial to understand how adults provide scaffolding to understand current practices (Lindeman, Jabot, & Berkley, 2014; Wang, Choi, Benson, Eggleston, & Weber, 2020), their possible impact on learning outcomes, the practicality of application in formal settings, and facilitate the execution of study replications. We faced problems to determine what the role of the adults (researchers or teachers) was during the activities, adult–child ratios, or how adults provided scaffolding, for example—whether there was a demonstration at the beginning of each activity or whether the adults asked questions to help guide the children. This gap in activities' reports was also observed by Angeli and Valanides (2020) that state, "they [reported efforts] do not [...] describe how teachers scaffolded young students' computational thinking".

Recommendation for reporting research: It is essential to have detailed information about how adults participated in the studies to facilitate comparisons between studies and provide scaffolding guidelines. We strongly recommend including this information in the methodology section when reporting. Moreover, it would be of interest for studies to explicitly include their rationale behind adults' roles during the activities. A previously conducted review study on the use of robotics to promote CT by Ioannou and Makridou (2018) mentions most of their examined studies promoted student-centered, constructivist learning; however, the authors also highlight the existence of reporting gaps and point out some aspects were not adequately presented in this area. Overall, our findings support their call for more accurate and specific reporting of educational practices.

3.3.6. Scaffolding

We categorized scaffolding into three categories: narrative, auxiliary objects, and embodied examples, assuming that teachers' support was always present and should not be a separate category. If a story motivated the programming task, we assigned it to the narrative category. If the study reported using objects, such as coding cards that represent commands or worksheets where the problem is solved before programming the robot, we associated the study with the auxiliary objects category. We marked the study as using embodied examples if children played games in which they imitated robots and their movements were "programmed" by their peers. However, these were not used as exclusive categories, as one study could include several types of scaffolding elements or strategies.

Regarding scaffolding strategies, the use of narratives to accompany the robotics activities was the most prevalent, followed by introducing auxiliary objects. The auxiliary objects included were coding cards, worksheets, symbolic annotations of commands, and mat models. All the studies with robots programmed with physical buttons used coding cards or symbolic commands' annotation as auxiliary objects.

Recommendation for future research: Future studies that use button-based programming interfaces should be aware that auxiliary objects, such as coding cards or commands' annotations, seem valuable resources that help preschool children while programming robots through physical buttons. As we pointed out in Section 3.2.1, physical button user interfaces do not provide visible code needed for debugging and iterations, so auxiliary materials should be included to mitigate this deficiency. In the case of button-based robots analyzed in this study, the Colby mouse robot includes coding cards (Code & Go Robot Mouse, n.d.) and in the case of Bee-Bot, coding cards are sold separately by the manufacturer (Command Card Set for Bee-Bot, n.d.).

3.3.7. Unplugged CT activities

We analyzed the activities' content to evaluate if they included unplugged activities aimed at engaging children in different modes of understanding CT concepts without using the robot. For example, using games that use arrow cards and worksheets where a trajectory is defined in paper or games in which one child plays a programmer role and uses a limited set of verbal commands to give directions to another child playing the robot. Six studies combined robot-mediated and unplugged activities to stimulate the development of CT. Only one study (Saxena et al., 2020) provided preliminary evidence that the unplugged activities could positively impact students' accomplishment in the plugged activities.

Recommendation for future research: In the studies where both robot-mediated and unplugged activities are present, it is difficult to state if CT improvement is associated only with robot-mediated activities or is also the result of unplugged activities. There should be more comparative studies to clarify this point as the "research to explore using both unplugged and plugged activities together for cultivating CT remains limited and undertheorized" (Saxena et al., 2020).

3.3.8. Explicit debugging

Debugging is widely considered as one of the principal components of CT (Shute et al., 2017), but the vast majority of the studies did not target this skill explicitly. Only two studies proposed activities to work on error detection and correction explicitly. Bers et al. (2014) worked with a curriculum that explicitly addressed debugging and Muñoz-Repiso and Caballero-González (2019) included activities in which children had to detect and correct programming errors.

Recommendation for CT curricula: We suppose that it is assumed that developing the code implies debugging. Although this might be true, evidence suggests debugging can be explicitly taught through strategies such as flowcharts or exercises with erroneous programs. A recent study with children found that including activities which specifically target this skill could have beneficial effects on learning. Wong and Jiang (2018) showed a Scratch based intervention was able to improve elementary school childrens' debugging skills through the presentation of pre-made errors in order to ensure sufficient opportunities to practice this skill. A literature review on debugging skills by Rich, Strickland, Binkowski, and Franklin (2019) suggests children as young as five years old are able to debug through trial and error practices but could achieve more sophisticated debugging strategies if provided with the necessary scaffolding and learning opportunities. As such, including explicit (pre-fixed) debugging opportunities in the curricula could prove more beneficial than the implicit debugging that naturally emerges from programming practices.

3.3.9. Communication and sharing

Three studies included activities aimed at communicating, sharing, and creating community. They were based on group discussions aimed at sharing ideas, strategies, and doubts.

Recommendation for future research: Activities that encourage children to relate their advances, share their experiences, and see others' works can help children reflect on their actions, see other approaches to solve the same problems, and advance in their knowledge building. In this sense, enabling sharing instances might contribute to advance in their zone of proximal development (Vygotsky, 1980) and enable them to learn new skills. The process of sharing ideas within a community allows children to compare their work to other projects (thus providing feedback) and build upon previous work (also known as remixing) (Kotsopoulos et al., 2017). In older users, this process is perhaps better visualized within the existing online Scratch programming communities. For example, Dasgupta, Hale, Monroy-Hernández, and Hill (2016) examined data from 1 million Scratch online users and found that those who practiced remixing more were exposed to a broader set of commands and were more likely to use them. Data from students' participation has also been explored. For example, a study by Fields, Giang, and Kafai (2014) used a random sample of 5 thousand users and found no significant association between community participation and sophistication in programming, with the exception of those who presented high levels of engagement. Roque, Rusk, and Resnick (2016) propose several ways in which creative and collaborative sharing can take place within these environments, such as organizing contests, focusing on niche user interests, or suggesting specific prompts. Altogether, these previous experiences with older users might provide researchers and practitioners with valuable hints to adapt these strategies for a younger audience. None of the studies evaluated the impact of social interactions aimed at sharing in knowledge acquisition. We consider that comparative studies that clarify this point would help to better design future CT curricula.

3.3.10. Teaching knowledge from other domains

Many studies included teaching knowledge from other domains through CT activities with robots. Proposed activities targeted self-expression through art and the development of imagination and creativity (Roussou & Rangoussi, 2019), music, dance, culture, language, mathematics and art (Bers et al., 2019), art, music and dance, facts about animals and safari environments (Pugnali et al., 2017), and recycling and household chores (Kazakoff et al., 2013). Art was a subject explored in almost all studies that integrated knowledge from domains other than programming and robotics.

Recommendation for future research: Although many different subjects were taught through activities that aimed to support CT development, we still do not know if there are areas more or less appropriate to be combined with CT activities. The only measured learning progress was the one related to CT, and it is not clear if the activities had an influence on the acquisition of knowledge in other domains and if there were domains that benefited more than the others. If there is a positive impact on learning, it should be analyzed in detail to provide guidelines that will help to integrate CT activities with other types of knowledge and concrete examples of activities that have demonstrated

Table 3
Types of assessments used in the reviewed studies.

Assessment	Description of categories	Reference
Portfolio	Evaluation of student's products during robotics activities through the use of rubrics or checklists	Angeli and Valanides (2020), Bers et al. (2014, 2019), Georgiou and Angeli (2019), González and Muñoz Repiso (2018), Khoo (2020), Muñoz-Repiso and Caballero-González (2019), Pugnali et al. (2017), Saxena et al. (2020), Sullivan and Bers (2013, 2016)
Traditional	Multiple choice and/or open ended tests evaluated by correctness and completeness and designed for summative purposes	Bers et al. (2019), Kazakoff et al. (2013), Nam et al. (2019), Roussou and Rangoussi (2019)
Interview	Structured conversation for qualitative data extraction	Bers et al. (2014), Sullivan and Bers (2013)
Survey	Self-reporting of skill level and/or confidence when performing tasks	Cho and Lee (2017)

positive results. We fully support Anwar et al.'s proposal that "more fine-grained studies are needed to understand the role of ER across contexts, activities, and disciplines for which they are best suited, and for what kind of students" and Yu and Roque's call for more support for teaching knowledge from other domains in future computational kits.

3.4. RQ3: How was computational thinking evaluated?

Most of the reviewed studies implemented ad-hoc evaluations or adaptations of ad-hoc evaluations created for previous studies (Angeli & Valanides, 2020; Bers et al., 2014, 2019; Cho & Lee, 2017; González & Muñoz Repiso, 2018; Khoo, 2020; Muñoz-Repiso & Caballero-González, 2019; Pugnali et al., 2017; Roussou & Rangoussi, 2019; Saxena et al., 2020; Sullivan & Bers, 2013, 2016), two of the studies implemented modified versions of Baron Cohen's picture sequencing test (Kazakoff et al., 2013; Nam et al., 2019).

Recommendation for future studies: The overwhelming use of ad-hoc evaluations and the lack of use of standard and validated assessments of CT hinders comparisons between approaches. We are aware of only two very recent CT assessments targeted towards preschool children, which report validity and reliability (Relkin, de Ruiter, & Bers, 2020; Zapata-Cáceres, Martín-Barroso, & Román-González, 2020). As valid and reliable tests begin to appear in the recent literature, further studies should implement these metrics.

3.4.1. Classification of assessments

We classified the assessments implemented into categories extracted from Tang and collaborators (Tang, Yin, Lin, Hadad, & Zhai, 2020), summarized in Table 3. Studies with multiple assessments were included more than once.

A majority of the reviewed studies use portfolio analysis to assess children's CT through the implementation of observational rubrics or checklists. This assessment method is probably preferred due to constraints given by participants' young age, in which reading and writing are in the process of being acquired skills, thus hindering the implementation of traditional tests that often rely on these skills for administration. This limitation could also apply to assessments based on surveys, which are heavily underrepresented in the selected studies (see Table 4).

Portfolio based-assessments for robotics activities, especially with robots that do not generate structured code, are costly regarding time and human resources, as each child needs to be individually assessed by a previously trained observer. Furthermore, training in scoring guidelines should be extensive to increase inter-observer reliability, which is underreported.

Finally, interviews appeared as an additional assessment option for portfolio analysis. However, how much the interviews influenced the evaluation was often unclear.

Recommendation for reporting research: Portfolio analysis appears as an easily accessible developmentally appropriate assessment; however, scoring guidelines need to be transparent and based on objective achievements to facilitate replication. We suggest supplementary information should include detailed descriptions of the implemented metrics, as sometimes the assessed elements are not sufficiently described (e.g., scoring based on how much help from adults the child needed to achieve a given task should include a description of how the different levels of help from adults were categorized). Similarly, less-structured elements within assessments, such as interviews used to obtain supplementary information, should be thoroughly described to replicate them. For several years, studies have been calling for further research on CT assessment (Lockwood & Mooney, 2017; Román-González, Moreno-León, & Robles, 2017a; Zhong, Wang, Chen, & Li, 2016). Recently, studies have reported growth in this area (Li et al., 2020; Tang et al., 2020), as assessment is an integral part of not only academic research but also CT's integration to educational settings. However, our results show CT assessments in preschoolers appear heterogeneous and so far no assessment tool has been adopted by a majority of studies.

3.4.2. Assessed concepts

Our examination of the reported concepts behind the assessments showed that sequencing ability was one of the central components of CT evaluations in 87% of the reviewed studies. Sequencing was measured by assessing children's robot-programming portfolios and traditional assessments such as picture sequencing tasks (Kazakoff et al., 2013; Roussou & Rangoussi, 2019). Debugging skills were reportedly assessed in 47% of publications, while algorithm design and pattern recognition were overwhelmingly less frequently assessed, appearing in just 15% of publications. Abstraction and decomposition were abilities mentioned in just 1 publication (8% of our sample) along with other practices such as hypothesis formulation and understanding cause and effect relations. Finally, 38% of the studies focused their assessment on learned programming concepts such as loops, conditionals, and 15% on sensor or numeric parameters.

Recommendation for future research: Perhaps due to the lack of theoretical consensus in the field in regards to CT (Angeli & Giannakos, 2020; Grover & Pea, 2013; Ioannou & Makridou, 2018; Shute et al., 2017), there is great variability in which concepts researchers targeted to assess their interventions. Existing trends in choosing to evaluate certain variables over others might be constrained by the availability of testing materials or developmental appropriateness to this particular age-range. As CT is an umbrella term, certain aspects of it might be more challenging to assess in young children. Further research on CT should expand on the concept's operationalization, as having a grasp on developmental trajectories for these concepts will allow researchers to create targeted interventions. Taking into account our findings

Table 4
Concept frequency in the reviewed publications.

Concept	Frequency	Percent	Reference
Sequencing	13	87%	Angeli and Valanides (2020), Bers et al. (2014, 2019), Georgiou and Angeli (2019), González and Muñoz Repiso (2018), Kazakoff et al. (2013), Muñoz-Repiso and Caballero-González (2019), Nam et al. (2019), Pugnali et al. (2017), Roussou and Rangoussi (2019), Saxena et al. (2020), Sullivan and Bers (2013, 2016)
Debugging	7	47%	Angeli and Valanides (2020), Bers et al. (2014, 2019), Khoo (2020), Muñoz-Repiso and Caballero-González (2019), Pugnali et al. (2017), Sullivan and Bers (2013)
Loops	5	38%	Bers et al. (2014, 2019), Pugnali et al. (2017), Sullivan and Bers (2013, 2016)
Conditionals	5	38%	Bers et al. (2014, 2019), Pugnali et al. (2017), Sullivan and Bers (2013, 2016)
Problem solving	2	15%	Nam et al. (2019), Roussou and Rangoussi (2019)
Numeric parameters	2	15%	Nam et al. (2019), Roussou and Rangoussi (2019)
Sensor parameters	2	15%	Nam et al. (2019), Roussou and Rangoussi (2019)
Action-instruction correspondence	2	15%	González and Muñoz Repiso (2018), Muñoz-Repiso and Caballero-González (2019)
Pattern recognition	2	15%	Khoo (2020), Saxena et al. (2020)
Algorithm design	2	15%	Khoo (2020), Saxena et al. (2020)
Coding/Programming	2	15%	Cho and Lee (2017), Khoo (2020)
Hypothesis formulation	1	8%	Roussou and Rangoussi (2019)
Cause and effect relations	1	8%	Roussou and Rangoussi (2019)
Efficiency	1	8%	Khoo (2020)
Decomposition	1	8%	Khoo (2020)
Abstraction	1	8%	Khoo (2020)

regarding both type of assessment and explored constructs, our overall results align with Tran's conclusion in regards to the current conflation between CT and programming concepts within assessments. This is evidenced by the prevalence of portfolio analysis based tests which evaluate performance on programming tasks (Lockwood & Mooney, 2017). Calls for the construction of valid and reliable assessments have been previously voiced by several researchers (Grover & Pea, 2013; Román-González et al., 2017a; Shute et al., 2017). While efforts have targeted mainly teenagers and adults (Kılıç, Gökoğlu, & Öztürk, 2020; Korkmaz, Çakir, & Özden, 2017; Kukul & Karatas, 2019; Román-González, Pérez-González, & Jiménez-Fernández, 2017b; Tsai, Liang, & Hsu, 2020), their contributions are relevant to the exploration of CT's possible factor structures and shed light on the contribution of different components towards our understanding of the CT concept.

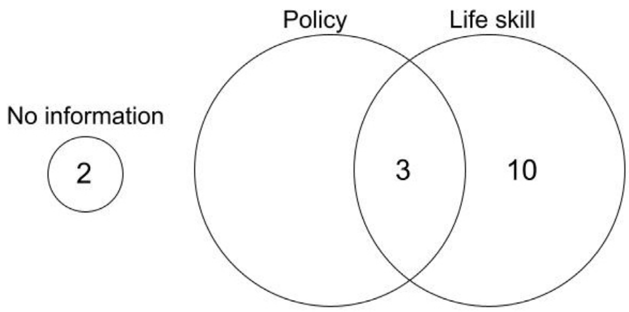


Fig. 2. Venn diagram of thematic overlap in the selected publications' research motivations.

3.4.3. Research designs

The research designs implemented were varied. A total of four studies (Kazakoff et al., 2013; Muñoz-Repiso & Caballero-González, 2019; Nam et al., 2019; Roussou & Rangoussi, 2019) used a control group and a pre-test and post-test design in order to test the effects of their robotics interventions. Eleven studies did not implement a control group. Of the latter, five studies (Bers et al., 2019; Cho & Lee, 2017; Pugnali et al., 2017; Saxena et al., 2020; Sullivan & Bers, 2016) implemented only a post-test evaluation, and six of them assessed their studies continuously or after each robotics session (Angeli & Valanides, 2020; Bers et al., 2014; Georgiou & Angeli, 2019; González & Muñoz Repiso, 2018; Khoo, 2020; Sullivan & Bers, 2013).

Recommendation for future studies: Further studies should continue to use experimental or quasi-experimental designs to test the effects of their interventions controlling for variables such as children's developmental outcomes (Steiner, Wroblewski, & Cook, 2009). This recommendation is especially relevant for studies that include a considerable sample size, as favorable results could point to effective and scalable evidence-based education practices. It is noteworthy that only one of the four reviewed studies, which included a larger sample size (over 100 children), included a control group in their design. Finally, longitudinal designs are recommended to test the permanence of any positive effects on children's outcomes and understand any novelty effects (Gustafsson, 2010). In this sense, our results are aligned with previous findings by Ioannou and Makridou (2018) on educational robotics for CT development, in which the authors provide a general overview of the variability found in their reviewed studies in regards to both research design and methodological approaches.

3.5. RQ4: Which individuals and countries are most active and influential in research on computational thinking development for preschool children mediated by robots, and what have been their motivations for conducting research in this area?

Analyzing author's motivations and rationale for studying CT and educational robotics in preschool children presented a challenging task. Authors did not always express their attitude towards statements, ideas, and visions they explicitly mentioned in the text. For this analysis, we assumed these explicit mentions included in their articles indicated agreement from its authors, and thus categorized the motivations based on this assumption.

3.5.1. Research motivation

Firstly, we categorized motivations based on Bocconi et al.'s article (Bocconi et al., 2016) on different rationales for including

Table 5
Research motivations presented by the authors.

Authors	Economic	Equity	Literacies	Citizenship	Education	Fulfillment
Angeli and Valanides (2020)			Yes		Yes	
Roussou and Rangoussi (2019)	Yes		Yes			
Khoo (2020)				Yes		
Nam et al. (2019)			Yes		Yes	
Bers et al. (2019)	Yes	Yes	Yes			
Georgiou and Angeli (2019)						
Muñoz-Repiso and Caballero-González (2019)			Yes	Yes	Yes	
González and Muñoz Repiso (2018)			Yes		Yes	Yes
Pugnali et al. (2017)			Yes			
Bers et al. (2014)			Yes		Yes	Yes
Saxena et al. (2020)				Yes		
Sullivan and Bers (2016)		Yes	Yes	Yes	Yes	
Kazakoff et al. (2013)	Yes	Yes	Yes	Yes	Yes	
Sullivan and Bers (2013)		Yes				
Cho and Lee (2017)			Yes	Yes		

CT in European curricula. Their first category was considering CT as a way to develop computer-related skills that are transferable to other domains, thus contributing to personal growth. These life skills should enable new ways of thinking and expressing and help individuals manage real-world situations and solve problems. The second category was based on boosting economic growth, filling job vacancies in ICT, and preparing the children for future job markets. Although these categories were defined based on the rationale for including CT in compulsory education, and not based on the motivations presented in research projects, they were the only resource focused on CT that we found that provides categories that can be applied in motivation analysis.

The most common motivation (see Fig. 2) in our context was the one related to life skills and it was mentioned in 13 publications (Angeli & Valanides, 2020; Bers et al., 2014, 2019; Cho & Lee, 2017; González & Muñoz Repiso, 2018; Kazakoff et al., 2013; Khoo, 2020; Muñoz-Repiso & Caballero-González, 2019; Nam et al., 2019; Pugnali et al., 2017; Roussou & Rangoussi, 2019; Saxena et al., 2020; Sullivan & Bers, 2016). This motivation was three times combined with arguments focused on economic policy (Bers et al., 2019; Kazakoff et al., 2013; Roussou & Rangoussi, 2019).

We were unable to classify two publications (Georgiou & Angeli, 2019; Sullivan & Bers, 2013) as the position of the authors to some statements lacked specificity (i.e., "The development of computational thinking is as important as writing, reading and arithmetic, and, it should start as early as kindergarten (Wing, 2008)." Georgiou and Angeli (2019)) or the work was motivated by gender disparity in STEM (science, technology, engineering, mathematics) fields (Sullivan & Bers, 2013), and this type of motivation was not included in the predetermined categories.

Secondly, to conduct a more detailed analysis of the rationale behind the conducted research, we classified our findings based on seven categories proposed by Vogel, Santo, and Ching (2017). They define the areas of impact present in arguments for universal computer science (CS) education: (1) economic and workforce development, (2) equity and social justice, (3) competencies and literacies, (4) citizenship and civic life, (5) scientific, technological and social innovation, (6) school improvement and reform and (7) fun, fulfillment and personal agency. Despite CT being considered a broader term than CS as it is not limited to computers and includes skills related to everyday activities and general problem solving (Shute et al., 2017), we considered that "competencies and literacies" category was able to encompass this "life skill" aspect of CT as it stresses the importance of a certain type of knowledge in supporting other important competencies. Moreover, this classification allowed us to provide a more nuanced perspective on authors' motivations.

Categories[3] found in each publication are presented in Table 5. The most common motivation was CT's applicability to real-life challenges and to support the development of knowledge from other domains reflected by the category "competencies and literacies". Authors stressed that "computational thinking skills are not skills that only computer scientists value, but, also skills that can be transferred to any domain, such as literacy, art, journalism, biology, engineering, mathematics, science, and many more" (Angeli & Valanides, 2020) and that robotics supports "transformation of abstract concepts of science, engineering, and technology into concrete real-world understanding" (Nam et al., 2019).

Motivations aimed towards school improvement were related to promoting students' engagement and introducing teachers to new tools and practices that enhance students' learning. Naturally, this type of motivation was often paired with the "competencies and literacies" category, which is more focused on student's individual outcomes but ultimately also refers to an aspect of the teaching–learning dyad. Authors observed the potential of the robots as a tool to support learning ("kindergarten educators have focused on robotics and computer programming as methods of teaching academic skills to kindergarteners through hands-on experiences with new technologies" (Nam et al., 2019), "robotics can provide a fun and playful way for teachers to integrate academic content with the creation of meaningful projects" (Bers et al., 2014)) and that "the current digital situation calls for the development of strategies to modernize learning processes" (Muñoz-Repiso & Caballero-González, 2019).

The third most frequent motivation was based on the idea of raising conscious citizens that understand the current digitalized world and building "a society of creators" (Vogel et al., 2017). This idea was mentioned, for example, by Sullivan and Bers (2016) who recognized that "While learning about the natural world is important, developing children's knowledge of the human-made world, the world of technology and engineering, is also needed for children to understand the environment they live in".

Aspects related to equity, and in our case related to gender equity, were mentioned in four articles, all co-authored by Marina Bers. They pointed out that children who are exposed to STEM curricula in childhood present fewer gender-based stereotypes regarding STEM careers (Bers et al., 2019; Kazakoff et al., 2013; Sullivan & Bers, 2016) and that early exposure to programming

[3] Economic stands for "economic and workforce development" category, equity for "equity and social justice", literacies for "competencies and literacies", citizenship for "citizenship and civic life", education for "school improvement and reform" and fulfillment for "fun, fulfillment and personal agency". Category "scientific, technological and social innovation" was not included in the table as it was not mentioned by any author.

and robotics helps to increase girls' interest in engineering fields before gender stereotypes are ingrained (Sullivan & Bers, 2013).

It was surprising that different motivations did not influence the structure or the types of activities used or the evaluation process.

Recommendation for future research: We observed that few studies offered a categorization and analysis of the motivations and rationales to prioritize CT in research, government, and economic policy. Categories that we used were either a very broad summary of reasons to include CT in education, not in research (Bocconi et al., 2016), or targeted CS, not CT (Vogel et al., 2017). An in-depth analysis of the motivations behind researching CT using robots in preschool and establishing categories for these motivations would be a valuable resource to enable a systematic analysis of current and future research.

3.6. Authors and geographic distribution

To better understand the collected data and provide a general overview of the research's geographical dimension, we analyzed the publications' authors and the countries in which the studies were conducted.

Our inclusion criteria emphasized empirical studies which explicitly target the concept of computational thinking through robotics in young children. As such, this meant our scope excluded studies in robotics involving concepts which might relate with CT, such as executive functions (Di Lieto et al., 2020), skills of scientific process (Turan & Aydoğdu, 2020) or creative thinking (Çakır, Korkmaz, İdil, & Erdoğmuş, 2021). Additionally, our focus on empirical studies with CT assessments may have led us to academics who implement a particular type of research design.

The most active researcher in this analysis is Marina Bers. She is a professor and researcher at Tufts University (USA), and she co-authored six publications. Her most frequent co-author, Amanda Sullivan (Tuft University, USA), participated in five of those articles. Both co-authored three of the most influential articles with over 200 citations (Bers et al. (2014) - 498 citations, Kazakoff et al. (2013) - 264 citations and Sullivan and Bers (2016) - 216 citations). Authors that participated in two publications are Elizabeth R. Kazakoff (co-author of Bers and Sullivan, USA), Charoula Angeli (Cyprus), Ana Muñoz-Repiso (Spain), and Yen-Air Caballero-González (Spain).

The research projects were conducted in Europe (Spain (Bers et al., 2019; González & Muñoz Repiso, 2018; Muñoz-Repiso & Caballero-González, 2019), Cyprus (Angeli & Valanides, 2020; Georgiou & Angeli, 2019), Greece (Roussou & Rangoussi, 2019)), North America (USA (Bers et al., 2014; Kazakoff et al., 2013; Pugnali et al., 2017; Sullivan & Bers, 2013, 2016)) and Asia (Republic of Korea (Cho & Lee, 2017; Nam et al., 2019), Hong Kong (Saxena et al., 2020), People's Republic of China (Khoo, 2020)). There were no studies from South America, Africa, or Australia.

Generally, the reviewed studies were heterogeneous in their reporting of socioeconomic and cultural characteristics of their samples. Participants' ethnic background was included in a few studies (Bers et al., 2014; Kazakoff et al., 2013; Sullivan & Bers, 2013, 2016). Studies set in formal education contexts reported school characteristics (i.e., public (Georgiou & Angeli, 2019; Kazakoff et al., 2013; Muñoz-Repiso & Caballero-González, 2019; Roussou & Rangoussi, 2019; Sullivan & Bers, 2016), private (Nam et al., 2019) or samples with both public and private schools (Bers et al., 2014, 2019; Sullivan & Bers, 2013)). A few studies did not report this information (Cho & Lee, 2017; González & Muñoz Repiso, 2018; Saxena et al., 2020). Angeli and Valanides (Angeli & Valanides, 2020) reported urban school settings, while Pugnali et al.'s participants were enrolled in a summer programme which required sign-up fees.

Recommendation for future research: As we mentioned in Section 3.3.3, aspects, such as, socioeconomic and cultural context are important factors to consider in the design of inclusive and effective interventions. Previous research demonstrated that successful educational programs that promote CT may need adaptations when implemented in different cultural and socioeconomic contexts (de Souza, Salgado, Leitão, & Serra, 2014) that is why we consider it beneficial for future research to conduct studies in a more geographically and culturally diverse context. Furthermore, as discussed in Section 3.5.1, several studies (see Table 5) argued learning CT might be helpful in preparing children to become digital citizens (i.e., promoting social and cultural participation through digital media) and foster equity (thus aiming to promote access and opportunity to all). In order to accomplish these goals, it stands to reason it is utterly important to be able to rely on evidence from diverse contexts in order to create tailored, culturally and contextually appropriate policies with effective results.

4. Limitations

Although we followed a systematic review process to provide objective and transparent results, the current study still has certain limitations. Our most notorious limitation is the sample size. Although we explored various databases and used inclusive search terms to obtain the greatest number of publications related to the subject, computational thinking is still quite a new concept, not particularly explored in early childhood education using robots, and there are not many empirical studies in this area. To increase the number of articles in this analysis, we decided to include marginal papers, sometimes turning a blind eye to the publication quality. We applied a backward snowballing approach (Kitchenham et al., 2015) and checked all publications cited in the papers included in our review. We also complemented the automated search with a manual search in Google Scholar, but all these approaches did not extend our scope.

Our sample included six publications co-authored by Marina Bers, which constitute $\frac{1}{3}$ of the total number of analyzed publications. This fact makes our results biased towards her approaches. It was impossible to overcome this bias since all the studies passed the selection process and are a relevant part of state of the art.

Not all of the extracted categories were explicitly reported in the original studies. For example, it was not always detailed if instances of free-play were part of the activities. In our results, we reported free-play only if it was clearly stated in the text. It does not ensure that publications that did not mention this activity actually did not have free-play instances. Our approach could drive to an incorrect interpretation of the absence of information. To overcome this difficulty and mitigate possible misinterpretations, we consulted additional information sources (online documentation, publications related to the same empirical study, complementary materials provided by the authors). However, it was not always possible to ensure that the absence of a direct report implied the absence of the variable we wanted to observe.

5. Discussion

Our overall results suggest using educational robotics to promote CT in early childhood constitutes a budding field that presents many opportunities for further research. The number of studies that completely met our inclusion criteria was low. Thus, it is reasonable to conclude that the use of ER to promote CT in early childhood, especially through systematic and assessed intervention, is a field that remains in its infancy and should continue to be explored (Khoo, 2020). Despite existing evidence suggesting robots are a feasible tool for promoting learning in young users (Jung & Hinds, 2018; Toh et al., 2016) and the growing interest in the inclusion of CT into the curriculum (Kakavas & Ugolini, 2019; Shute et al., 2017) few studies have attempted to evaluate ER interventions for CT enhancement in early childhood systematically.

The studies analyzed provided many valuable results that contribute to the necessary body of knowledge required for scalable and impactful interventions aimed at enhancing young children's computational thinking, exploring the possibility for developmentally appropriate interventions, and raising interesting questions regarding best practices for bringing robots and CT into early childhood classrooms.

In this sense, our results have contributed to identifying current trends regarding research involving robotics to promote CT in early childhood with systematic assessments and thus could become a helpful guideline for both practitioners and researchers interested in this field. Furthermore, our results allowed us to recognize existing knowledge gaps and limitations within the field, thus contributing to the identification of further inquiry lines for both researchers and practitioners.

5.1. Reporting gaps

Our analysis of ER activities for promoting CT identified several reporting gaps that could impact intervention effectiveness. Specifically, factors such as interventions' duration and frequency, groups' size, and detailing the role and participation of adults throughout the intervention, which could have a high impact on replication and feasibility, were found to be significantly underreported. It is especially relevant from a policy-making point of view (e.g., assessing the cost-effectiveness of the interventions and its requirements regarding human capital) as these elements could significantly impact the overall effectiveness of the experience. Previous studies have found that learning gains could differ based on whether group-based approaches or individual approaches are taken (Zhong & Li, 2020) and depending on the presence of scaffolding elements and adults' roles (Wang et al., 2020). In addition, learning practitioners would benefit from more detailed reporting since understanding the context in which specific learning goals were achieved would help with the classroom orchestration of CT activities (Ioannou & Makridou, 2018). To contribute to closing these reporting gaps, we propose a rubric that could be used to summarize the activities developed to stimulate CT (see Appendix B).

5.2. Improvements in CT evaluation

Our exploration of the assessments utilized in the studies targeting preschoolers was congruent with recent literature on CT assessments (Tang et al., 2020) regarding diversity in assessed constructs and underreporting of instruments' validity and reliability. Recent efforts have been made to create valid, reliable, and developmentally appropriate assessments for this particular life-stage (Relkin et al., 2020; Zapata-Cáceres et al., 2020). Furthermore, these assessments are advantageous because they would allow measures independent from the devices used for the intervention. Accompanying this, further studies should build upon the still few but increasing efforts to include experimental or quasi-experimental designs, as previous studies have pointed out in regards to educational experiences with the use of robots (Anwar et al., 2019).

5.3. Opportunities in robot design

Regarding the robots utilized, we observed little variability in the studied tools, with half of the studies implementing robotics programming through physical buttons and providing a low level of embodiment required for the task. As different robots might provide different learning opportunities, further comparative studies should be conducted to identify the strengths and weaknesses of the available options and best practices for the developmentally appropriate design of ER tools for early childhood. Aspects like input interface, feedback provided by the robot, and programmable robot actions should be investigated in-depth so that future design decisions can be grounded on empirical data and theoretical knowledge. Furthermore, we identified widespread use of commercially available robots and a lack of low-cost open-source robots targeting early childhood education that could be easily adopted by teachers and researchers. Existing open projects (Open-Source Robotics Hardware Projects, n.d.) do not provide a stable robotic platform that could be easily used in a school context by users without programming knowledge. We understand that using commercially available, stable platforms can be a good starting point, but it should be noticed that we need more research and community to support free ER platforms that target early childhood. Open-source and open-hardware tools would enable researchers to adapt the robot's capabilities and programming interface to diverse research conditions while also providing flexibility and transparency when collecting data. It would also ensure that the robots' characteristics and capabilities are appropriate for educational contexts and not designed with the primary goal of entertainment. Finally, such solutions would provide greater stability to projects that would not depend on companies to continue producing a specific product. Such stability could help with scaling up efforts and broader impacts.

5.4. Designing CT learning experiences

Finally, our summary of activities' content and structure (see Section 3.3) describes current intervention practices that could be useful for curriculum development. Further research in this area will contribute to establishing evidence-based curricula and classroom orchestration guidelines that seem to be in demand (Ioannou & Makridou, 2018). For example, creating a broad set of CT activities that are incrementally challenging and examining their learning outcomes could be a useful resource for educators and researchers alike. Additionally, educational activities that target early childhood are typically embedded in games and playful scenarios connected with storytelling (Hirsh-Pasek et al., 2009). Thus, the presentation dynamics related to motivational factors should be a central aspect of this research area. Moreover, the impact of gender, disabilities, socioeconomic and cultural context, and generational differences should be considered when designing and evaluating these activities.

Overall, the previously stated considerations seem particularly important since many studies were motivated by the existing economic and government policies that acknowledge the importance of incorporating CT into educational programs and official curricula (Bocconi et al., 2016; Uscanga et al., 2019). These efforts have been varied. Regardless, most have been characterized by the imperative requirement of interdisciplinarity to incorporate technological solutions that are developmentally appropriate and provide meaningful learning opportunities. Our results contribute to these objectives by encompassing both educational and technological aspects of early childhood interventions, which should be useful when considering future large scale implementations of ER to promote CT.

6. Conclusion

This review conducted a systematic analysis of the ER interventions and experiences to promote CT in early childhood. We examined 15 empirical studies. We presented results that reflect the types of robots utilized, the main characteristics of the proposed ER activities, and the different ways CT was assessed. Lastly, we presented the most influential research groups within the examined literature and their stated motivations for conducting their research. Our main objective was to conduct a broad exploration and analysis of the reported experiences with ER to promote CT in early childhood and examine existing gaps.

The results obtained indicate that all the studies used commercial robots, in most cases, with limited capacities of input and output interfaces and aspects that could be improved upon to increase developmental appropriateness to children's cognitive level. We observed a lack of consensus about how to implement CT-related activities — there was significant variability in duration, structure, and content. The CT evaluation process was also very heterogeneous, considering the type of metrics, assessed concepts, and research design. The diverse approaches could not be explained by the study's motivation or by their geographical location.

Together, our findings point to a nascent research area with many opportunities to explore technologies, activities, and social contexts. Our findings suggest a need for this area of study to mature through more rigorous reporting of research experiences and consistent approaches to evaluate CT. We proposed a rubric that should enable consistent and comprehensive reporting of activities (see Appendix B). Finally, future research should also take into account practical aspects such as activity group sizes in order to make it more likely that these activities could be implemented at scale.

As the importance of CT is recognized by more and more governments and educational agencies, there is a need to develop concrete technological solutions that allow large scale implementation of ER activities aimed at promoting CT in early childhood education. The scientific community should support the implementation of these activities with theoretically informed tools and evidence-based guidelines.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This review is a part of a Ph.D. project of the first author that was supported by Comisión Académica de Posgrado (Uruguay). We would also like to thank Sebastián Pizard for his guidance during the systematic review process.

Appendix A. Extracted fields

We used the following fields in the extraction form. It was possible to select more than one option in the case of option-based questions.

1. Publication details
    - Authors (String)
    - Title (String)
    - Year (Number)
    - Source title (String)
    - DOI (String URL)
    - Abstract (String)
2. Robot
    - Name (String)
    - Cost (Number, value in USD)
    - Available input interfaces (Options: physical buttons, tangibles, graphical, hybrid, other)
    - Output interfaces (Options: light, sound, movement, other)
    - Availability (Options: opensource, commercial)
3. Activities
    - Context and structure
        - Context (Options: formal, informal)
        - Modality of work (Options: individual, group-based, mixed)
        - Type of activity (Options: goal-oriented, open-ended, mixed)
        - Free-play (Options: yes, no)
        - Activities total duration (Time in hours)
        - Session duration (Time in minutes)
        - Session frequency (Number per week)
    - Adults participation
        - Adult–child ratio (Number adult/Number child)
        - Adult scaffolding (String with description of actions)
    - Scaffolding
        - Narrative (Options: yes, no)
        - Auxiliary objects (Options: yes, no)
        - Embodied examples (Options: yes, no)
    - Content
        - Unplugged CT (Options: yes, no)
        - Explicit debugging (Options: yes, no)
        - Teaching content not related to CT (String: list of areas)
        - Communicating, sharing, and creating community (Options: yes, no)
4. Evaluation
    - Metric type (Options: ad-hoc, adapted ad-hoc, validated)
    - Assessment type (Options: portfolio, traditional, interview, survey)
    - Assessed concepts (String: list of concepts)
    - Group comparison (Options: none, control group, between conditions)
    - Number of evaluations (Number)
    - Moment of evaluation (Options: pre, post, during)

Appendix B. Rubric to report activities

We identified the following items that we consider valuable to report:

- Activity context (Options: formal, informal)
- Number of participants (Number)
- Modality of work (Options: individual, group-based, mixed)
    - Number of participants per group (Number)
- Type of activity (Options: goal-oriented, open-ended, mixed)
- Free-play (Options: yes, no)

Table C.6
Search engine, search term, date of the search execution, number of items found and additional information.

Engine	Search term	Note	Date	Nr of items
SCOPUS	TITLE-ABS-KEY (((robot* AND "computational thinking" AND ((preschool*) OR ("young children") OR ("early age*") OR (kindergarten*) OR ("lower education") OR (childhood) OR ("early years") OR ("elementary education") OR ("young learner*")))))	Search in title, abstract and keywords.	03.08.20	62
IEEE XPLORE	"All Metadata":"computational thinking" AND "All Metadata":"robot*" AND ("All Metadata":"preschool*" OR "All Metadata":"young children" OR "All Metadata":"early age*" OR "All Metadata":"kindergarten*" OR "All Metadata":"lower education" OR "All Metadata":"childhood" OR "All Metadata":"early years" OR "All Metadata":"elementary education" OR "All Metadata":"young learner*")	Search in all metadata.	31.07.20	9
ScienceDirect	Term 1: robot AND "computational thinking" AND ((preschool) OR ("young children") OR ("early age") OR (kindergarten) OR ("lower education") OR (childhood) OR ("early years")) Term 2: robot AND "computational thinking" AND (("elementary education") OR ("young learner"))	Search in title, abstract and keywords. ScienceDirect allows maximum 8 boolean terms in the search term, so we split the search term in 2 to cover all the words identified as relevant for the search.	03.08.20	4 and 3
SpringerLink	robot* AND "computational thinking" AND ((preschool*) OR ("young children") OR ("early age*") OR (kindergarten*) OR ("lower education") OR (childhood) OR ("early years") OR ("elementary education") OR ("young learner*"))	SpringerLink does not allow to restrict the search to title, abstract and keywords so the search was done in the whole article.	22.07.20	172
ACM	Abstract:(robot*) AND Abstract:("computational thinking") AND Abstract:("preschool*" OR "young children" OR "early age*" OR "kindergarten*" OR "lower education" OR "childhood" OR "early years" OR "elementary education" OR "young learner*")	Search in abstract.	31.07.20	6

Table C.7
Summary of publication details of the selected studies and information about robot used, age range and number of children at target age.

Authors	Title	Year	Robot	Nr	Age
Angeli and Valanides (2020)	Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy	2020	Bee-Bot	50	5 to 6
Roussou and Rangoussi (2019)	On the use of robotics for the development of computational thinking in kindergarten: Educational intervention and evaluation	2020	Colby mouse	20	4.5 to 6
Khoo (2020)	A case study on how children develop computational thinking collaboratively with robotics toys	2019	Colby Mouse and Ozobot Bit	3	5
Nam et al. (2019)	Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners	2019	TurtleBot	53	5 to 6
Bers et al. (2019)	Coding as a playground: Promoting positive learning experiences in childhood classrooms	2019	KIBO	172	3 to 5
Georgiou and Angeli (2019)	Developing preschool children's computational thinking with educational robotics: The role of cognitive differences and scaffolding	2019	Bee-Bot	180	5 to 6
Muñoz-Repiso and Caballero-González (2019)	Robotics to develop computational thinking in early Childhood Education [Robótica para desarrollar el pensamiento computacional en Educación Infantil]	2019	Bee-Bot	131	3 to 6
González and Muñoz Repiso (2018)	A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education	2018	Bee-Bot	131	3 to 6
Pugnali et al. (2017)	The impact of user interface on young children's computational thinking	2017	KIBO	11	4 to 7
Bers et al. (2014)	Computational thinking and tinkering: Exploration of an early childhood robotics curriculum	2014	LEGO	53	4.9 to 6.5
Saxena et al. (2020)	Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education	2020	Bee-Bot	11	3 to 6
Sullivan and Bers (2016)	Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade	2016	KIWI	33	4 to 8
Kazakoff et al. (2013)	The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood	2013	LEGO wedo	42	4 to 6
Sullivan and Bers (2013)	Gender differences in kindergarteners' robotics and programming achievement	2013	LEGO mindstorm RIS	53	5 to 6
Cho and Lee (2017)	Possibility of improving computational thinking through activity based learning strategy for young children	2017	LEGO mindstorm NXT	12	5 to 6

- Activity duration
  - Number of sessions (Number)
  - Session duration (Number in minutes)
  - Sessions' frequency (Number per week)
- Adults guiding the activities
  - Number (Number)
  - Adult–child ratio (Number)
- Scaffolding provided
  - Adults' support (String with description of the type and degree of support)
  - Narrative (String with description)
  - Auxiliary objects (String with description)
  - Embodied examples (String with description)
  - Others (String with description)
- Unplugged activities (String with description)
- Explicit error detection (Options: yes, no)
- Relation with other domains (String with description)
- Communicating, sharing, and creating community (Options: yes, no)

Appendix C. Complementary material

See Tables C.6 and C.7.

References

Ahnert, J., Schneider, W., & Bös, K. (2010). Developmental changes and individual stability of motor abilities from the preschool period to young adulthood. In *Human development from early childhood to early adulthood* (pp. 45–72). Psychology Press.

Almjally, A., Howland, K., & Good, J. (2020). Investigating children's spontaneous gestures when programming using TUIs and GUIs. In *Proceedings of the interaction design and children conference* (pp. 36–48). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3392063.3394408, https://doi.org/10.1145/3392063.3394408.

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, Article 106185.

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, Article 105954.

Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, *9*(2), http://dx.doi.org/10.7771/2157-9288.1223.

Bakała, E., Visca, J., Tejera, G., Seré, A., Amorin, G., & Gómez-Sena, L. (2019). Designing child-robot interaction with robotito. In *2019 28th IEEE international conference on robot and human interactive communication* (pp. 1–6). IEEE.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20–23.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, *58*(3), 978–988. http://dx.doi.org/10.1016/j.compedu.2011.10.006, http://dx.doi.org/10.1016/j.compedu.2011.10.006.

Bers, M. U. (2010). The tangiblek robotics program: Applied computational thinking for young children.. *Early Childhood Research & Practice*, *12*(2), n2.

Bers, M. U. (2018). Coding and computational thinking in early childhood: the impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8.

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, *72*, 145–157. http://dx.doi.org/10.1016/j.compedu.2013.10.020.

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., et al. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice*: *Tech. rep.*, Joint Research Centre (Seville site).

Bornstein, M. H., Hahn, C.-S., & Suwalsky, J. T. (2013). Physically developed and exploratory young infants contribute to their own long-term academic achievement. *Psychological Science*, *24*(10), 1906–1917.

Botički, I., Kovačević, P., Pivalica, D., & Seow, P. (2018). Identifying patterns in computational thinking problem solving in early primary education. In *Proceedings of the 26th international conference on computers in education* (p. 6).

Campbell, F., Conti, G., Heckman, J. J., Moon, S. H., Pinto, R., Pungello, E., et al. (2014). Early childhood investments substantially boost adult health. *Science*, *343*(6178), 1478–1485.

Çakır, R., Korkmaz, Ö., İdil, Ö., & Erdoğmuş, F. U. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. *Thinking Skills and Creativity*, *40*, Article 100812.

Cho, Y., & Lee, Y. (2017). Possibility of improving computational thinking through activity based learning strategy for young children. *Journal of Theoretical and Applied Information Technology*, *95*(18), 4385–4393.

Code & Go Robot Mouse. (n.d.). URL https://www.learningresources.com/stem-robot-mouse.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Command Card Set for Bee-Bot. (n.d.). URL https://www.terrapinlogo.com/products/bee-blue-accessories/bee-bot-command-cards.html.

Cuny, J., Snyder, L., & Wing, J. (2010). Computational thinking: a definition,

Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). Remixing as a pathway to computational thinking. In *proceedings of the 19th acm conference on computer-supported cooperative work & social computing* (pp. 1438–1449).

de Souza, C. S., Salgado, L. C., Leitão, C. F., & Serra, M. M. (2014). Cultural appropriation of computational thinking acquisition research: Seeding fields of diversity. In *Proceedings of the 2014 conference on innovation & technology in computer science education* (pp. 117–122). New York, NY, USA: Association for Computing Machinery, https://doi.org/10.1145/2591708.2591729.

Di Lieto, M. C., Pecini, C., Castro, E., Inguaggiato, E., Cecchi, F., Dario, P., et al. (2020). Empowering executive functions in 5-and 6-year-old typically developing children through educational robotics: An RCT study. *Frontiers in Psychology*, *10*, 3084.

DiCarlo, C. F., Pierce, S. H., Baumgartner, J., Harris, M. E., & Ota, C. (2012). Whole-group instruction practices and children's attention: A preliminary report. *Journal of Research in Childhood Education*, *26*(2), 154–168.

Doyle, O., Harmon, C. P., Heckman, J. J., & Tremblay, R. E. (2009). Investing in early human development: timing and economic efficiency. *Economics & Human Biology*, *7*(1), 1–6.

Fields, D. A., Giang, M., & Kafai, Y. (2014). Programming in the wild: trends in youth computational participation in the online scratch community. In *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 2–11).

Frei, P., Su, V., Mikhak, B., & Ishii, H. (2000). <i>curlybot</i>: Designing a new class of computational toys. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 129–136). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/332040.332416, https://doi.org/10.1145/332040.332416.

Georgiou, K., & Angeli, C. (2019). Developing preschool children's computational thinking with educational robotics: The role of cognitive differences and scaffolding. In *16th international conference on cognition and exploratory learning in digital age* (pp. 101–108). http://dx.doi.org/10.33965/celda2019_201911l013.

Gerosa, A., Koleszar, V., Gómez-Sena, L., Tejera, G., & Carboni, A. (2019). Educational robotics and computational thinking development in preschool. In *2019 XIV Latin American conference on learning technologies* (pp. 226–230). IEEE.

González, Y. A. C., & Muñoz Repiso, A. G.-V. (2018). A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education. In *Proceedings of the sixth international conference on technological ecosystems for enhancing multiculturality* (pp. 41–45). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3284179.3284188, https://doi.org/10.1145/3284179.3284188.

Grant, M. J., & Booth, A. (2009). A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, *26*(2), 91–108.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Gustafsson, J.-E. (2010). Longitudinal designs. In *Methodological advances in educational effectiveness research* (pp. 91–115). Routledge.

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2020). An emerging technology report on computational toys in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213–224.

Heckman, J., & Carneiro, P. (2003). *Human capital policy*: *Tech. rep.*, National Bureau of Economic Research.

Hirsh-Pasek, K., Golinkoff, R. M., Berk, L. E., Singer, D., et al. (2009). *A mandate for playful learning in preschool: Applying the scientific evidence*. Oxford University Press.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*(September 2017), 296–310. http://dx.doi.org/10.1016/j.compedu.2018.07.004, https://doi.org/10.1016/j.compedu.2018.07.004.

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. http://dx.doi.org/10.1007/s10639-018-9729-z.

Isnaini, R., & Budiyanto, C. (2018). The influence of educational robotics to computational thinking skill in early childhood education. In *Conference: The 1st international conference on computer science and engineering technology at: Kudus, Indonesia* (pp. 617–626). http://dx.doi.org/10.4108/eai.24-10-2018.2280538.

Jung, M., & Hinds, P. (2018). Robots in the wild: A time for more robust theories of human-robot interaction. *Journal of Human-Robotics Interaction*, *7*(1), http://dx.doi.org/10.1145/3208975, https://doi.org/10.1145/3208975.

Jung, S. E., & Won, E. S. (2018a). Systematic review of research trends in robotics education for young children. *Sustainability (Switzerland)*, *10*(4), 1–24. http://dx.doi.org/10.3390/su10040905.

Jung, S. E., & Won, E.-s. (2018b). Systematic review of research trends in robotics education for young children. *Sustainability*, *10*(4), 905.

Kakavas, P., & Ugolini, F. C. (2019). Computational thinking in primary education: a systematic literature review. *Research on Education and Media*, *11*(2), 64–94.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*(4), 245–255. http://dx.doi.org/10.1007/s10643-012-0554-5.

Khoo, K. Y. (2020). A case study on how children develop computational thinking collaboratively with robotics toys. *International Journal of Educational Technology and Learning*, *9*(1), 39–51.

Kılıç, S., Gökoğlu, S., & Öztürk, M. (2020). A valid and reliable scale for developing programming-oriented computational thinking. *Journal of Educational Computing Research*, Article 0735633120964402.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews (Vol. 4)*. CRC press.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., et al. (2010). Systematic literature reviews in software engineering-a tertiary study. *Information and Software Technology*, *52*(8), 792–805. http://dx.doi.org/10.1016/j.infsof.2010.03.006, http://dx.doi.org/10.1016/j.infsof.2010.03.006.

Komis, V., Romero, M., & Misirli, A. (2016). A scenario-based approach for designing educational robotics activities for co-creative problem solving. In *International conference edurobotics 2016* (pp. 158–169). Springer.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569.

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., et al. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171.

Kubilinskiene, S., Zilinskiene, I., Dagiene, V., & Sinkevičius, V. (2017). Applying robotics in school education: a systematic review. *Baltic Journal of Modern Computing*, *5*(1), 50–69. http://dx.doi.org/10.22364/bjmc.2017.5.1.04.

Kukul, V., & Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity and reliability. *Informatics in Education*, *18*(1), 151–164.

Lee, R. L. T., Lane, S. J., Tang, A. C. Y., Leung, C., Louie, L. H. T., Browne, G., et al. (2020). Effects of an unstructured free play and mindfulness intervention on wellbeing in kindergarten students. *International Journal of Environmental Research and Public Health*, *17*(15), 5382.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., et al. (2020). On computational thinking and STEM education. *Journal for STEM Education Research*, *3*(2), 147–166. http://dx.doi.org/10.1007/s41979-020-00044-w, https://doi.org/10.1007/s41979-020-00044-w.

Lindeman, K. W., Jabot, M., & Berkley, M. T. (2014). The role of STEM (or STEAM) in the early childhood setting. In *Learning across the early childhood curriculum* (pp. 95–114). Emerald Group Publishing Limited.

Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? A systematic literary review. ArXiv preprint arXiv:1703.07659.

Lozada, M., & Carro, N. (2016). Embodied action improves cognition in children: evidence from a study based on piagetian conservation tasks. *Frontiers in Psychology*, *7*, 393.

Macedonia, M., Müller, K., & Friederici, A. (2011). The impact of iconic gestures on foreign language word learning and its neural substrate. *Human Brain Mapping*, *32*.

McLean, M., & Harlow, D. (2017). Designing inclusive STEM activities: A comparison of playful interactive experiences across gender. In *Proceedings of the 2017 conference on interaction design and children* (pp. 567–574). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3078072.3084326, https://doi.org/10.1145/3078072.3084326.

Merkouris, A., & Chorianopoulos, K. (2019). Programming embodied interactions with a remotely controlled educational robot. *ACM Transactions on Computing Education (TOCE)*, *19*(4), 1–19.

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G., the PRISMA Group (2009). Reprint—Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Physical Therapy*, *89*(9), 873–880. http://dx.doi.org/10.1093/ptj/89.9.873.

Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early childhood education. *Comunicar*, *27*(59), 63–72. http://dx.doi.org/10.3916/C59-2019-06.

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting plans to action: The effects of a card-coded robotics curriculum and activities on Korean kindergartners. *The Asia-Pacific Education Researcher*, *28*(5), 387–397. http://dx.doi.org/10.1007/s40299-019-00438-4, https://doi.org/10.1007/s40299-019-00438-4.

Nicholson, M. (2019). *Preschool students learning from play based learning and small group instruction* (Master's thesis), Northwestern College.

Oberer, N., Gashaj, V., & Roebers, C. M. (2017). Motor skills in kindergarten: internal structure, cognitive correlates and relationships to background variables. *Human Movement Science*, *52*, 170–180.

One Laptop Per Child. (n.d.). URL https://www.onelaptopperchild.org/.

Open-Source Robotics Hardware Projects. (n.d.). URL https://www.eng.yale.edu/grablab/openrobothardware.html.

Plan Ceibal. (n.d.). URL https://www.ceibal.edu.uy/en/institucional.

Poncela, A., & Gallardo-Estrella, L. (2015). Command-based voice teleoperation of a mobile robot via a human-robot interface. *Robotica*, *33*(1), 1.

Pons, P., & Jaen, J. (2019). Interactive spaces for children: gesture elicitation for controlling ground mini-robots. *Journal of Ambient Intelligence and Humanized Computing*, 1–22.

Pugnali, A., Sullivan, A., & Bers, M. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*(1), 171–193.

Raffle, H. S., Parkes, A. J., & Ishii, H. (2004). Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 647–654).

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). Techcheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*(4), 482–498. http://dx.doi.org/10.1007/s10956-020-09831-x, https://doi.org/10.1007/s10956-020-09831-x.

Resnick, M., & Rusk, N. (2020). Coding at a crossroads. *Communications of the ACM*, *63*(11), 120–127. http://dx.doi.org/10.1145/3375546.

Rhemtulla, M., & Tucker-Drob, E. M. (2011). Correlated longitudinal changes across linguistic, achievement, and psychomotor domains in early childhood: evidence for a global dimension of development. *Developmental Science*, *14*(5), 1245–1254.

Rich, K. M., Strickland, C., Binkowski, T. A., & Franklin, D. (2019). A k-8 debugging learning trajectory derived from research literature. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 745–751).

Román-González, M., Moreno-León, J., & Robles, G. (2017a). Complementary tools for computational thinking assessment. In S. C. Kong, J. Sheldon, & K. Y. Li (Eds.), *Proceedings of international conference on computational thinking education* (pp. 154–159). the Education University of Hong Kong.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017b). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678–691.

Roque, R., Rusk, N., & Resnick, M. (2016). Supporting diverse and creative collaboration in the scratch online community. In *Mass collaboration and education* (pp. 241–256). Springer.

Roussou, E., & Rangoussi, M. (2019). On the use of robotics for the development of computational thinking in kindergarten: educational intervention and evaluation. In *International conference on robotics and education RiE 2017* (pp. 31–44). Springer.

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158.

Silvis, D., Lee, V. R., Clarke-Midura, J., Shumway, J., & Kozlowski, J. (2020). Blending everyday movement and representational infrastructure: An interaction analysis of kindergarteners coding robot routes. In *The interdisciplinarity of the learning sciences, 14th international conference of the learning sciences* (pp. 98–105).

So, H.-J., Jong, M. S.-Y., & Liu, C.-C. (2020). Computational thinking education in the Asian Pacific region. *The Asia-Pacific Education Researcher*, *29*, 1–8.

Steiner, P. M., Wroblewski, A., & Cook, T. D. (2009). Randomized experiments and quasi-experimental designs in educational research. *The Sage Handbook of Educational Evaluation*, 75–96.

Sullivan, J. V. (2018). Learning and embodied cognition: A review and proposal. *Psychology Learning & Teaching*, *17*(2), 128–143.

Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education*, *23*(3), 691–702.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20.

Superti Pantoja, L., Diederich, K., Crawford, L., & Hourcade, J. P. (2019). Explorations of voice user interfaces for 3 to 4 year old children. In *Extended abstracts of the 2019 chi conference on human factors in computing systems* (pp. 1–6).

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, *148*(July), http://dx.doi.org/10.1016/j.compedu.2019.103798.

Tejera, G., Amorin, G., Sere, A., Capricho, N., Margenat, P., & Visca, J. (2019). Robotito: programming robots from preschool to undergraduate school level. In *2019 19th international conference on advanced robotics* (pp. 296–301). IEEE.

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M., & Yeo, S. H. (2016). A review on the use of robots in education and young children. *Educational Technology and Society*, *19*(2), 148–163.

Tran, Y. (2018). Computer programming effects in elementary: Perceptions and career aspirations in STEM. *Technology, Knowledge and Learning*, *23*(2), 273–299.

Tsai, M.-J., Liang, J.-C., & Hsu, C.-Y. (2020). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, Article 0735633120972356.

Tsujimoto, S. (2008). The prefrontal cortex: Functional neural development during early childhood. *The Neuroscientist*, *14*(4), 345–358.

Turan, S., & Aydoğdu, F. (2020). Effect of coding and robotic education on pre-school children's skills of scientific process. *Education and Information Technologies*, *25*(5), 4353–4363.

Umam, M. U. K., Budiyanto, C., & Rahmawati, A. (2019). Literature review of robotics learning devices to facilitate the development of computational thinking in early childhood. In *AIP conference proceedings (Vol. 2194)*. AIP Publishing LLC.

Uscanga, E. A. V., Bottamedi, J., & Brizuela, M. L. (2019). Pensamiento computacional en el aula: el desafío en los sistemas educativos de Latinoamérica. *Revista Interuniversitaria de Investigación en Tecnología Educativa*.

van der Fels, I. M., te Wierike, S. C., Hartman, E., Elferink-Gemser, M. T., Smith, J., & Visscher, C. (2015). The relationship between motor skills and cognitive skills in 4–16 year old typically developing children: A systematic review. *Journal of Science and Medicine in Sport*, *18*(6), 697–703.

Vogel, S., Santo, R., & Ching, D. (2017). Visions of computer science education: Unpacking arguments for and projected impacts of CS4all initiatives. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 609–614). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3017680.3017755, https://doi.org/10.1145/3017680.3017755.

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2020). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development*, 1–23.

Weintrop, D., & Wilensky, U. (2018). How block-based, text-based, and hybrid block/text modalities shape novice programming practices. *International Journal of Child-Computer Interaction*, *17*(July), 83–92. http://dx.doi.org/10.1016/j.ijcci.2018.04.005, https://doi.org/10.1016/j.ijcci.2018.04.005.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *24*(3), 33. http://dx.doi.org/10.1145/1118178.1118215.

Wong, G. K., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging. In *2018 IEEE international conference on teaching, assessment, and learning for engineering* (pp. 328–334). IEEE.

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

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In *2020 IEEE global engineering education conference* (pp. 1905–1914). IEEE.

Zhong, B., & Li, T. (2020). Can pair learning improve students' troubleshooting performance in robotics education? *Journal of Educational Computing Research*, *58*(1), 220–248.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, *53*(4), 562–590.

Zhu, K., Ma, X., Wong, G. K. W., & Huen, J. M. H. (2016). How different input and output modalities support coding as a problem-solving process for children. In *Proceedings of the the 15th international conference on interaction design and children* (pp. 238–245).

Appendix 3

Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten

Ewelina Bakala et al. «Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten». In: *Proceedings of the 21st annual acm interaction design and children conference.* 2022, pp. 422–429

Author's contribution The author conceptualized the research idea, designed the methodology, and developed the data collection strategies. She participated in the field study and, alongside three other researchers, conducted the content analysis. She also led and actively participated in the writing process.

Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten

Ewelina Bakala
ebakala@fing.edu.uy
Instituto de Computación,
Universidad de la República
Montevideo, Uruguay

Anaclara Gerosa
agerosa@psico.edu.uy
CICEA, Universidad de la República
Montevideo, Uruguay

Juan Pablo Hourcade
juanpablo-hourcade@uiowa.edu
Department of Computer Science,
The University of Iowa
Iowa, USA

María Pascale
maria9pascale@gmail.com
Escuela Universitaria Centro de
Diseño, Universidad de la República
Montevideo, Uruguay

Camila Hergatacorzian
cam.hergata@gmail.com
Escuela Universitaria Centro de
Diseño, Universidad de la República
Montevideo, Uruguay

Gonzalo Tejera
gtejera@fing.edu.uy
Instituto de Computación,
Universidad de la República
Montevideo, Uruguay

Figure 1: From left to right, Botley, Robotito, and Blue-Bot.

ABSTRACT

Programmable robots designed for preliterate children are one of the options being explored and put into practice for teaching computational thinking skills to children in preschool and kindergarten. Classroom use of these robots may involve use by groups of children due to cost, logistical, and pedagogical reasons. To understand design factors affecting the social use of these robots, we explored the use of three programmable robots with distinctive design characteristics in a kindergarten classroom. Our findings suggest that programmable robot designs that may work well for use by individual children may cause difficulties when shared by groups of children if not all children in the group are able to easily perceive the input (program), output (robot actions), or program state. Based on these design factors we provide recommendations for the design of programmable robots, their evaluation for social use, and for addressing design limitations with support by adult facilitators.

IDC '22, June 27–30, 2022, Braga, Portugal

ACM ISBN 978-1-4503-9197-9/22/06…$15.00
https://doi.org/10.1145/3501712.3529745

CCS CONCEPTS

• **Applied computing** → **Collaborative learning**; • **Human-centered computing**; • **Social and professional topics** → Children;

KEYWORDS

kindergarten children, computational thinking, programmable robot, social use, perception

ACM Reference Format:
Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, María Pascale, Camila Hergatacorzian, and Gonzalo Tejera . 2022. Design Factors Affecting the Social Use of Programmable Robots to Learn Computational Thinking in Kindergarten. In *Interaction Design and Children (IDC '22), June 27–30, 2022, Braga, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3501712.3529745

1 INTRODUCTION

Robots have long been an attractive option for teaching computational thinking concepts [59, 60] to children, going as far back as early efforts enabling children to control a robotic turtle through the LOGO programming language [23]. The use of robots for computational thinking education has not relented since then [4, 10, 37, 49, 61]. Authors have proposed that robots, as physical objects, allow children to learn in an embodied way promoting their motor skills [6]. Moreover, robots are similar to toys and other items children encounter daily, thus easing their transition to programming through relatable and practical hands-on experiences [37]. As scholars, educators, and policy makers explore teaching computational thinking concepts in kindergarten settings with preliterate children [7], there is a need to consider the social aspects of programmable robots designed for early childhood education. Not doing so could make the use of robots for teaching computational thinking impractical in these settings, in particular in lower-income regions of the world, as they would likely have to be shared by groups of children.

In this short paper we describe our experience exploring the social aspects of three programmable robots designed for early childhood, each featuring distinctive user interfaces, by conducting the same activities with each of them in a kindergarten classroom with preliterate 5-6-year-old children in a lower middle-income urban setting in Uruguay. Through our analysis of video from study sessions we identify key design factors affecting social use of the robots. These design factors can be used to motivate the design of new programmable robots, understand areas where groups of children will need extra support to address design limitations, and evaluate programmable robots for this age group.

In the remainder of the paper, we discuss related work on socially oriented programming for young children, provide details on our methods, present the design factors we identified together with suggestions for addressing design limitations, and discuss research and practical implications.

2 RELATED WORK

Social aspects of computational thinking education are an important consideration for preliterate children in schools, where logistics, costs, and accepted practices would in most cases prevent a one-kit-per-child approach. For example, in the metropolitan area where we conducted this study, a typical kindergarten classroom has about 24 children, with the typical school having two or three kindergarten groups. The cost of purchasing one programmable robot per child would be prohibitive. In addition, the logistics of storing and maintaining (e.g., charging) one robot per child without any extra staff, space, or infrastructure would make working with them impractical in most schools similar to the one where we conducted the research. There are also pedagogical reasons, with kindergarten curricula often emphasizing social aspects of learning (e.g., [58]), which have also been recognized by the child-computer interaction community (see [21], Chapters 2 and 8).

Given the emphasis on social aspects of learning in early childhood education, it is not surprising that researchers have studied a variety of programming environments for young children that could support groups of children working together, including room environments and tangible kits. For example, Montemayor et al. studied the design of programmable rooms, where 4-6-year-old children could program a physical environment by associating sensor events (e.g., a physical button being pressed) with an action by an actuator (e.g., a sound coming out of a speaker) using tangible tools [31]. A more studied approach is the use of tangible blocks for programming, which has included Horn's work on museum exhibits, which were designed for social aspects typical of a museum environment [19, 20]. Perhaps the closest efforts to those in our research were those with the Tangicons system, which in its first two versions enabled children in kindergarten to use blocks to program digital lights [47, 48]. The researchers behind Tangicons designed the system specifically for groups of children to work together, negotiating how to set up instructions [47].

Other work that has studied social aspects of computational thinking activities with this age group has focused on how activities are organized, rather than on the characteristics of kits. For example, Lee et al. found that during a summer camp activity for kindergarten children in which each child used their own robotic kit, children were more likely to engage socially with peers when using an unstructured robotics curriculum rather than a structured one [25]. The same research group, with a similar setup, compared learning computational thinking with a robotic kit versus a popular app, assessing social aspects, but in a context in which each child had their own kit or tablet [52]. In a similar manner, Fessakis et al. focused on the impact on social aspects of a pedagogical approach to using a programming environment with kindergarten children, rather than on how the design of the programming environment affected social dynamics [14]. Likewise, Caballero-Gonzalez et al. studied collaborative aspects of kindergarten children's computational thinking activities with a specific programmable robot [8]. Other similar examples include those of Roussou and Rangoussi [45] and Mantzanidou [28].

To the best of our knowledge, none of these efforts studied how different kits or kit characteristics affected the social dynamics of computational thinking activities as they happen in kindergarten classrooms. In fact, in recent literature reviews on computational thinking activities for young children, Papadakis notes the lack of support for social aspects in commonly used apps [38], while McCormick and Hall note the importance of incorporating social aspects [30] while pointing at only one study that discussed collaborative aspects between children while using one kit [32].

3 RESEARCH SETUP

Our research objective was to identify design factors in programmable robot kits that affect their social use by groups of children in kindergarten. More specifically, we were interested in social aspects as children are using the robots, rather than social aspects that may occur afterward, such as sharing outcomes with peers.

3.1 Participants

15 preschoolers (8 girls and 7 boys between the ages of 5 and 6) participated in the study (see Section 8 for recruitment details). Children attended level 5 (kindergarten) at a public school in Montevideo, Uruguay and belonged to the same classroom within the school. Convenience sampling was implemented. Socioeconomic levels for the participating institution have been characterized as

low-middle according to the country's public education authorities. 14 of the participating children presented neurotypical development, according to the children's teacher. Informed consent was obtained from each participant's parent or caregiver.

3.2 Materials and Procedure

We selected three robots for our activities (see Figure 1), Blue-Bot [55], Botley [41], and Robotito [5, 53], because these robots feature three different programming interfaces. All three share basic capabilities involving motion on a flat surface while not incorporating features such as programmable arms, legs, sounds, and so forth. The three robots' capabilities are typical for programmable robots designed for this age group.

Blue-Bot (15 cm x 15 cm x 8.2 cm) [55] is a bee-shaped robot with buttons integrated on its back. The buttons are used to introduce a sequence of robot actions (available options are move forward, move backward, turn right, turn left and pause), to clear the stored program, or to start the execution of the stored sequence. Similar programming interfaces can be found in educational robots that target young learners, such as, Bee-Bot [54] and Pro-Bot [56] (robots from the same company that manufactures Blue-Bot), Code & Go Robot Mouse [42], Robot Mind Designer [11], and VEX 123 [57]. Bee-Bot has been featured in many studies of computational thinking activities with preliterate children in preschool or kindergarten [3, 8, 16, 33, 46].

Botley (20.6 cm x 15.7 cm x 15.7 cm) [41] is a robot controlled via remote control that allows users to store and execute a sequence of movements similar to those of Blue-Bot: move forward, backward, turn right, and turn left. It also provides buttons to clear and execute the stored program. It has additional buttons for programming behaviors related to obstacle detection, loops, and volume control that were not used in our activities. We selected Botley to see if having the controls separate from the robot in a handheld device made a difference in group dynamics. Other robots, such as VEX 123 [57], also have the option of inputting programs from a handheld device separate from the robot, such as a smartphone.

In the case of Robotito (16.5 cm of diameter x 7.2 cm) [5, 53], children can program it by placing color patches on the floor. Each color is associated with one direction (left, right, forward, or backward), displayed on the corresponding side of the top of the robot with lights. A given light briefly shines brighter if Robotito just read an instruction corresponding to that light, color, and direction. The idea of sensing cards placed below the robot is present in educational robots such as Sphero Indi [51], KUBO [24], and Qobo [43]. We selected Robotito because programming is separate from the robot and distributed across the floor, presenting a different way of programming from the button-based interfaces in Blue-Bot and Botley.

To conduct activities with the robots, children worked in groups of five, with each group working with a different robot. Groups of five children have previously been identified as optimal by preschool teachers for collaborative activities [58]. We conducted the activities in a separate classroom within the school grounds. Each of the activities lasted approximately 25 minutes and were held in succession. The same member of the research team worked with each of the groups to coordinate the activities.

Children worked on mats of approximately 1 x 1 meters. Due to the different step lengths of Blue-Bot and Botley we used two different mats. We used flags that came with Botley (see Figure 3) to indicate the start and end position of the robot on the mat. The activities for each of the robots were analogous. One facilitator introduced children to how to program each robot and asked children to perform simple tasks. The tasks were borrowed from Di Lieto et al.'s work, from which we included tasks 1 (move forward), 2 (move backward), and 3 (L-shape to the left) [12] while adding tasks asking the children to make the robot go two spaces to the left and two spaces to the right between tasks 2 and 3. At the end of each activity, a facilitator asked children which aspects of the robot they enjoyed, whether they would add any features to it, whether they had found activities to be too hard or too easy, and what they liked the least about the experience. Lastly, a facilitator asked children to explain how the robot worked and how could they programmed it to grasp their overall understanding of it.

3.3 Data collection and analysis

We filmed each group using two digital video cameras, one located on the side of the mat to capture children's expressions from the front, and one located such that it could capture activities from above and get an overall view of the entire mat and children's distribution. A total of four researchers were present during the activities, with one coordinating the activities with the children and three conducting observations and taking notes throughout the process.

We conducted a conventional content analysis [22] of the video recordings to identify themes related to social aspects of the programmable robots. All four researchers involved in the sessions watched all video recordings and wrote open-ended observations using Google Jamboard's virtual sticky notes [17]. We then transcribed the 134 sticky notes into Lucidchart [26], due to its greater ease of manipulation, where three researchers, one of whom did not participate in the sessions, organized the sticky notes into themes and subthemes over several meetings.

4 RESULTS AND RECOMMENDATIONS

The theme that most consistently appeared as we organized observations was a socially oriented version of Norman's concept of visibility [34, 35]. Norman called for designers to make visible the state and available actions of devices. In other words, when perceiving a device, users should know what it is doing and what they can do with it [34, 35]. While Norman's focus was mainly on the personal use of devices, in our case what mattered was the use of a device (a programmable robot), by a group of children. The types of group perception that mattered in our case were: perceiving the input (i.e., the program), perceiving the output (i.e., seeing the robot execute a program), and perceiving the program state (knowing what instruction was being executed). Two related design issues were those of positional perspective and sensory distractors. We expand on all these below and include recommendations related to each theme.

4.1 Perception of Input

We found that enabling the entire group of children to perceive the commands entered during programming promoted collaboration between children through pointing and verbalizations. During the activities, one child typically programmed the sequences and others took on the role of observers and overseers of the task. Children who were able to see their classmate's actions or the created program were able to provide helpful feedback and suggest corrections to the program, thus improving the group's overall performance and learning. They contributed to program design by adding comments like "put it here!" (Robotito), "press this button now" (Botley) or by pointing to a button and saying: "you missed this" (Blue-Bot).

We observed that it was often difficult for children to see what commands their classmates entered when using Blue-Bot or Botley, as their buttons are small. Moreover, children tended to enter several commands quickly, requiring children to use their working memory to retain the program, thus causing them to commit errors which would not occur otherwise, while making it very difficult for their group mates to provide feedback. In other words, programmable robot designs that do not enable groups of children to perceive a program get in the way of groups of children discussing options, thinking about alternatives, and learning together. Botley worked better than Blue-Bot though because its remote-control made it easier for children to take turns while Blue-Bot activities tended to be more chaotic, with multiple children attempting to operate it simultaneously (see Figure 2).

Figure 2: Three children gathering around Blue-Bot, preventing two others from clearly seeing the commands being entered.

On the other hand, Robotito's interface and its use of colored cards allowed all children to easily perceive the program they were creating, as it was laid out on the floor mat where the robot moved (see Figure 4). As a consequence, this setup enabled all children to suggest any necessary adjustments and communicate with each other about them. For example, while seeing a program, a girl said to the group "I have an idea!" and proceeded to move a colored patch to a different location to correct the program.

4.1.1 Recommendation: Ensure Program Instructions are Easy to Perceive by All Children.

Some approaches may make it easier for groups of children to perceive program instructions. Robotito's instructions, for example, are color patches laid on the surface where the robot operates, which Robotito reads as it moves around. Other approaches include the use of tangible elements like cards, tiles, or blocks [1, 9, 27, 29, 36, 40]. All of these could involve problems with accessibility for at least some children with vision impairments. Designers should add additional supports to ensure accessibility, such as adding sound or tactile components to instructions. Having input controls on the robot itself could cause problems with perceiving input, although these could possibly be alleviated by making the program being entered accessible in other ways (e.g., instructions read out loud, instructions displayed on a screen). If the design of a programmable robot makes it difficult for groups of children to perceive program instructions, facilitators should structure activities such that they enable a discussion of programming options before entering instructions. Some researchers, for example, have used cards that represent instructions to enable children to carefully consider programs before inputting them into a robot [3, 8, 16, 33, 45, 46], and such an approach would also help address group perception of the program.

Figure 3: One child entering commands for Botley, with input from another child, with three other children unable to see what commands are being entered.

4.2 Perception of Output

The ability for groups of children to perceive the output of their programs has a direct impact on how well they can discuss whether, how, and why the program worked. Using a physical programming robot could therefore be advantageous for group learning experiences when compared to programming virtual robots on a screen, which depending on its size, may not enable as many children to perceive the execution of a program.

All the robots we used had the potential to have visible output to all participating children. However, we identified issues with Blue-Bot, due to having its controls incorporated on its back. This design resulted in some cases in children following the robot while it was moving to touch the programming buttons or stretching their arms around the robot to protect it from the other group members and prevent them from interacting with it. These behaviors made it such that some children in the group could not see what the robot was doing (see Figure 2). The other two robots, with their control separate from the robot itself, did not have this problem.

4.2.1 Recommendation: Ensure Robot Behavior as It Executes Programs is Easy to Perceive by All Children. From a robot-design perspective the first component of this recommendation would involve the robot being large enough to be seen by the group of children working with it. While such an approach is likely to accommodate many children with vision impairments, additional supports should be used to ensure that blind children would be able to perceive output [2, 39, 44]. Likewise, design features that could attract children to gather on top of the robot, such as having inviting inputs on their back, should be avoided. If using a robot that does not naturally enable perception of output by all participating children, facilitators should plan to add scaffolds to the activity to enable it.

Figure 4: An example of an activity with Robotito, with color patches used to program it.

4.3 Perception of Program State

What we mean by perception of program state is enabling children to perceive what instruction the robot is currently executing. The concept of program state perception is related to another classic human-computer interaction concept brought up by Norman, that of feedback [34, 35]. Good feedback entails users seeing the consequence of an action, in other words, understanding how an action affects the state of a device, enabling them to make a connection between selecting an action and its effect on a system [34, 35]. In the context of programming, understanding the effect of individual instructions is a common debugging approach (i.e., going through instructions step-by-step). Out of the robots we used, Blue-Bot did not have any features that would enable children to perceive its program state, Botley had a set of lights corresponding to instructions on its back that briefly lit up as it executed instructions, while Robotito, by reading instructions on the surface on which it moved and turning lights corresponding to instructions, made it very transparent as to what instructions it had already executed, what instruction it was currently executing, and which instructions were left to execute. In our observations, children did not appear to make a connection between Botley's lights and instructions being executed. We also did not tell them why the lights lit up. With Robotito, seeing the program being executed in the context of prior, current, and future instructions enabled children to discuss and correct programs as they were running and understand where a program failed in case it did not work as intended.

4.3.1 Recommendation: Enable Perception of Current Program State in the Context of Prior and Future Instructions. Designers of programmable robots should include features that enable children to perceive what instruction is currently being executed, in the context of knowing which instructions have already been executed, and which ones are coming up. Ideally, these features should be combined with either a speed of execution that is slow enough for children to keep track of instructions and their effect, as with Robotito, or having a way of controlling when the next instruction should be executed (e.g., [50]). As with other features, accessibility should be incorporated into the system's design. When using systems without these features, facilitators could accomplish a similar outcome by laying out an entire program using cards or other tangible materials corresponding to instructions, and then having children enter and run one instruction at a time.

Figure 5: A child realizing that Botley turned in the wrong direction (due to children entering commands from a perspective opposite to that of the robot), gesticulating toward the correct direction.

4.4 Positional Perspective

Children experience an exponential development in their mental rotation skills during preschool years [15] and effective training of these skills has been carried out for children in this age group [13]. Children's initial spatial and mental rotation skills might mediate their ability to successfully implement programs which require the robot's rotation to be completed. During our observations, we found that a robot which did not require children to mentally rotate their perspective such as Robotito was more intuitive and allowed children to complete the tasks more easily. We found that children committed more orientation mistakes for robots which required rotation such as Blue-Bot and Botley and were more successful in the tasks when their own orientation coincided with that of the robot.

For example, during our session with Botley, we asked children to complete an L shaped sequence which required the robot to move forward twice and then twice to the right. The programming child (A), who was sitting in front of the robot made a rotation mistake in his program with the aid of a classmate (B) who was sitting next to him in the same position with respect to the robot (the mistake made was pressing the left button instead of right). During the execution of the program, child B realized their mistake, pointing towards the right direction (see Figure 5) and verbally exclaiming for the robot to move in the opposite direction. Afterwards, he corrected the robot's position by manually rotating it to face the objective. With Blue-Bot, we found that children often did not make rotation mistakes in the first few instructions, since they would tend to position themselves in the same position as the robot to program the sequences.

4.4.1 Recommendation: Consider Visual Perspective for Robot Tasks Requiring Rotation.

It may be helpful for children who are beginners in programming tasks involving rotation to always take a perspective which coincides with the robot's and for teachers and facilitators to introduce rotation gradually as children develop these spatial skills. In these beginner stages, having children take multiple perspectives could potentially cause confusion, frustration, or "cheating" (e.g., children manually rotating the robot). If learning mental rotation skills is not a goal for the activities, a programmable robot like Robotito, in which perspective does not matter, may be preferrable.

4.5 Sensory Distractors

We noticed that some auxiliary materials, such as flags, were a source of distraction for children. Even though children enjoyed having access to these objects, they tended to hinder the group's ability to fulfil their objectives, as children would often move them before completing the program. They were also seen as another object to play with and attracted children's attention when they were supposed to be focused on programming or on the researcher who led the activities. Similarly, children seemed to find the buttons on Blue-Bot very inviting to press, getting a thrill out of pressing them without thinking of whether or how the button presses would impact the robot's behavior.

4.5.1 Recommendation: Avoid Adding Features to Kits or Activities that Can Distract Children.

Designers should be careful not to introduce features that could distract children from activity goals.

4.6 Individual Initiative and Group Learning

Individual initiative was beneficial for group learning, as children observed peers try instructions, and made their own suggestions to help peers as they considered what to do next. On other occasions, the more curious children explored robot parts trying to understand how they worked and so, made other group members discover things they probably wouldn't have discovered on their own. For example, when two children found Robotito's color sensor, another group member that was observing from a distance, came closer to look at it. In our observations, these individual initiatives helped with group learning, benefiting shy children.

4.6.1 Recommendation: Encourage Individual Initiative and Exploration Combined with Feedback and Sharing with the Rest of the Group.

Designs should invite exploration and enable quick recognition and recovery from mistakes. In addition, facilitators should encourage children's exploration and sharing with peers.

5 DISCUSSION

Our findings suggest that programmable robot designs that may work well for use by individual children may cause difficulties when shared by groups of children if not all children in the group are able to easily perceive the input (program), output (robot actions), or program state. In addition, extra sensory features that may support activities with individual children could lead to distractions in a group setting. Many of the commercially available programmable robots for early childhood appear to have been designed for individual use, mainly in homes, rather than for use in classrooms, with groups of children. While usability at the individual level is still very important, for robots that will be used in groups, the additional design factors we identified also matter.

Researchers designing programmable robots for use in early childhood classrooms should consider the design factors we identified to lower barriers to adoption in schools that would want to use them with groups of children. From a practitioner's perspective, those interested in implementing computational thinking activities in kindergarten with groups of children using programmable robots can use the design factors we identified to evaluate programmable robot kits for their suitability. At the same time, we identified, in our recommendations, additional support that could be provided by adult facilitators to address design shortcomings and enable group perception of input, output, and program state.

6 LIMITATIONS

The first limitation of our work is that we could have included more robotic kits. For example, many projects have used kits that include the use of physical blocks for programming, but we did not have access to one of these programmable robot kits. However, we are confident that the factors we identified in our results would still be relevant to these other kits. The second limitation is that children interacted with these programmable robots for a limited amount of time. It is possible that social patterns of use could change over time, but the basic advantages and limitations with respect to the design of the robots and their impact on social awareness of input, system state, and output, are unlikely to change. Finally, we conducted the study with a small group of children. However, issues of group perception, for example, are unlikely to change with other groups.

7 CONCLUSION

In a kindergarten setting, computational thinking activities with programmable robots may often involve groups of children working together with one robot due to logistical, financial, and pedagogical reasons. To better understand design factors that affect the social use of programmable robots in a kindergarten classroom we observed children in groups of five use three programmable robots with a variety of design characteristics. The design factors we identified from these observations pointed at the importance of all children in a group being able to perceive the input (i.e., the program), the output (i.e., robot actions), and program state (knowing what instruction was being executed). Two related design issues were those of positional perspective and sensory distractors. Based on these observations, we provided recommendations for both the design of programmable robots for group use and for facilitators implementing additional supports to address design limitations. We expect our contribution will help inform the design of future programmable robots for this age group and help practitioners better understand factors they should consider when selecting programmable robots and preparing for educational activities with them.

8 SELECTION AND PARTICIPATION OF CHILDREN

The study protocol used was approved by the ethics board of the lead institution. We obtained informed consent from parents or caregivers and were invited to conduct research by the preschool’s Director and the children’s teacher. Children’s parents were provided with detailed information about the study in writing and were encouraged to contact the main researcher if they had any questions about it. We invited children to participate in our activities, but children decided how engaged (or not) they wanted to be. If a child did not wish to participate in the activities, they could continue with their regular activities in the classroom as usual. We followed all pandemic related policies by education and health authorities. We conducted the study at a time of very low COVID19 infection rates. All methods were in accordance with the Declaration of Helsinki [18].

ACKNOWLEDGMENTS

We thank the Computational Thinking department in Plan Ceibal, Uruguay for providing robots used in the present study.

REFERENCES

[1] 2022. Plobot. Retrieved April 29, 2022 from https://plobot.com/
[2] 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 (ASSETS ’20)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3373625.3418003 event-place: Virtual Event, Greece.
[3] Charoula Angeli and Nicos Valanides. 2020. Developing young children’s computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior* 105 (April 2020), 105954. https://doi.org/10.1016/j.chb.2019.03.018
[4] Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, and Gonzalo Tejera. 2021. Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction* 29 (Sept. 2021), 100337. https://doi.org/10.1016/j.ijcci.2021.100337
[5] Ewelina Bakala, Jorge Visca, Gonzalo Tejera, Andrés Seré, Guillermo Amorin, and Leonel Gómez-Sena. 2019. Designing child-robot interaction with Robotito. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 1–6.
[6] Marina Umaschi Bers. 2017. *Coding as a playground: Programming and computational thinking in the early childhood classroom.* Routledge.
[7] Stefania Bocconi, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, Katja Engelhardt, and others. 2016. *Developing computational thinking in compulsory education-Implications for policy and practice.* Technical Report. Joint Research Centre (Seville site).
[8] Yen-Air Caballero-Gonzalez, Ana García-Valcárcel Muñoz-Repiso, and Alicia García-Holgado. 2019. Learning Computational Thinking and Social Skills Development in Young Children through Problem Solving with Educational Robotics. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM’19)*. Association for Computing Machinery, New York, NY, USA, 19–23. https://doi.org/10.1145/3362789.3362874 event-place: León, Spain.
[9] Kunal Chawla, Megan Chiou, Alfredo Sandes, and Paulo Blikstein. 2013. Dr. Wagon: A ’stretchable’ Toolkit for Tangible Computer Programming. In *Proceedings of the 12th International Conference on Interaction Design and Children (IDC ’13)*. Association for Computing Machinery, New York, NY, USA, 561–564. https://doi.org/10.1145/2485760.2485865 event-place: New York, New York, USA.
[10] Yu-Hui Ching, Yu-Chang Hsu, and Sally Baldwin. 2018. Developing Computational Thinking with Educational Technologies for Young Learners. *TechTrends* 62, 6 (Nov. 2018), 563–573. https://doi.org/10.1007/s11528-018-0292-7
[11] Clementoni. 2022. Robot Mind Designer. Retrieved April 29, 2022 from https://www.clementoni.com/pl-en/50534-robot-mind-designer/
[12] Maria Chiara Di Lieto, Emanuela Inguaggiato, Emanuela Castro, Francesca Cecchi, Giovanni Cioni, Marta Dell’Omo, Cecilia Laschi, Chiara Pecini, Giacomo Santerini, Giuseppina Sgandurra, and others. 2017. Educational Robotics intervention on Executive Functions in preschool children: A pilot study. *Computers in human behavior* 71 (2017), 16–23. Publisher: Elsevier.
[13] Laura M Fernández-Méndez, María José Contreras, and M Rosa Elosúa. 2018. From what age is mental rotation training effective? Differences in preschool age but not in sex. *Frontiers in Psychology* 9 (2018), 753. Publisher: Frontiers.
[14] G. Fessakis, E. Gouli, and E. Mavroudi. 2013. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education* 63 (April 2013), 87–97. https://doi.org/10.1016/j.compedu.2012.11.016
[15] Andrea Frick, Melissa A Hansen, and Nora S Newcombe. 2013. Development of mental rotation in 3-to 5-year-old children. *Cognitive Development* 28, 4 (2013), 386–399. Publisher: Elsevier.
[16] Kyriakoula Georgiou and Charoula Angeli. 2019. Developing Preschool Children’s Computational Thinking with Educational Robotics: The Role of Cognitive Differences and Scaffolding. In *Proceedings of the 16th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2019)*. IADIS Press, 101–108. https://doi.org/10.33965/celda2019_201911L013
[17] Google. 2022. Google Jamboard. Retrieved April 29, 2022 from https://jamboard.google.com/
[18] WEU-Do Helsinki. 2013. World medical association declaration of Helsinki. *Fortaleza, Brazil* (2013).
[19] Michael S. Horn, Erin Treacy Solovey, R. Jordan Crouser, and Robert J.K. Jacob. 2009. Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’09)*. Association for Computing Machinery, New York, NY, USA, 975–984. https://doi.org/10.1145/1518701.1518851 event-place: Boston, MA, USA.
[20] Michael S. Horn, Erin Treacy Solovey, and Robert J. K. Jacob. 2008. Tangible Programming and Informal Science Learning: Making TUIs Work for Museums. In *Proceedings of the 7th International Conference on Interaction Design and Children (IDC ’08)*. Association for Computing Machinery, New York, NY, USA, 194–201. https://doi.org/10.1145/1463689.1463756 event-place: Chicago, Illinois.
[21] Juan Pablo Hourcade. 2015. *Child-Computer Interaction.* Self, Iowa City, Iowa.
[22] Hsiu-Fang Hsieh and Sarah E. Shannon. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research* 15, 9 (Nov. 2005), 1277–1288. https://doi.org/10.1177/1049732305276687 Publisher: SAGE Publications Inc.
[23] Walter J Koetke. 1973. Supertoys, A New Approach to Learning Mathematics.
[24] KUBO. 2022. Discover how easy it can be to teach computer science | KUBO. Retrieved April 29, 2022 from https://kubo.education/
[25] Kenneth T. H. Lee, Amanda Sullivan, and Marina U. Bers. 2013. Collaboration by Design: Using Robotics to Foster Social Interaction in Kindergarten. *Computers in the Schools* 30, 3 (July 2013), 271–281. https://doi.org/10.1080/07380569.2013.805676 Publisher: Routledge.
[26] Lucid. 2022. Intelligent Diagramming. Retrieved April 29, 2022 from https://www.lucidchart.com
[27] makeblock. 2022. mTiny - An Early Childhood Education Robot for Kids. Retrieved April 29, 2022 from https://www.makeblock.com/mtiny
[28] Garyfalia Mantzanidou. 2019. Educational robotics in kindergarten, a case study. In *International Conference on Robotics in Education (RiE)*. Springer, 52–58.
[29] Matatalab. 2022. Home | matatalab. Retrieved April 29, 2022 from https://matatalab.com/en
[30] Kate I. McCormick and Jacob A. Hall. 2021. Computational thinking learning experiences, outcomes, and research in preschool settings: a scoping review of literature. *Education and Information Technologies* (Oct. 2021). https://doi.org/10.1007/s10639-021-10765-z
[31] Jaime Montemayor, Allison Druin, Gene Chipman, Allison Farber, and Mona Leigh Guha. 2004. Tools for Children to Create Physical Interactive Storyrooms. *Comput. Entertain.* 2, 1 (Jan. 2004), 12–12. https://doi.org/10.1145/973801.973821
[32] Karen Janette Murcia and Kok-Sing Tang. 2019. Exploring the multimodality of young children’s coding. *Australian Educational Computing* 34, 1 (2019).
[33] Ana Muñoz-Repiso and Yen-Air Caballero-González. 2019. Robotics to develop computational thinking in early Childhood Education. *Comunicar* 27, 59 (April

2019), 63–72. https://doi.org/10.3916/C59-2019-06
[34] Don Norman. 2013. *The design of everyday things: Revised and expanded edition.* Basic books.
[35] Donald A Norman. 1988. *The psychology of everyday things.* Basic books.
[36] obotics. 2022. Robot Kits For Kids | KIBO | KinderLab Robotics. Retrieved April 29, 2022 from https://kinderlabrobotics.com/kibo/
[37] Stamatios Papadakis. 2020. Robots and Robotics Kits for Early Childhood and First School Age. *International Journal of Interactive Mobile Technologies (iJIM)* 14, 18 (Nov. 2020), 34. https://doi.org/10.3991/ijim.v14i18.16631
[38] Stamatis Papadakis. 2021. The Impact of Coding Apps to Support Young Children in Computational Thinking and Computational Fluency. A Literature Review. *Frontiers in Education* 6 (2021), 183. https://doi.org/10.3389/feduc.2021.657895
[39] 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 event-place: London, United Kingdom.
[40] Primo. 2022. Meet Cubetto - Primo Toys Cubetto: A toy robot teaching kids code & computer programming. Retrieved April 29, 2022 from https://www.primotoys.com/
[41] Learning Resources. 2022. Botley. Retrieved April 29, 2022 from https://www.learningresources.com/botleyr-the-coding-robot-activity-set
[42] Learning Resources. 2022. Code & Go. Retrieved April 29, 2022 from https://www.learningresources.com/code-gor-robot-mouse-activity-set
[43] Robobloq. 2022. Qobo - Robobloq Co. Ltd. Retrieved April 29, 2022 from https://www.robobloq.com/product/Qobo
[44] Filipa Rocha, Guilherme Guimarães, David Gonçalves, Ana Cristina Pires, Lúcia Verónica Abreu, and Tiago Guerreiro. 2021. Fostering Collaboration with Asymmetric Roles in Accessible Programming Environments for Children with Mixed-Visual-Abilities. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3441852.3476553
[45] Evgenia Roussou and Maria Rangoussi. 2019. On the use of robotics for the development of computational thinking in kindergarten: Educational intervention and evaluation. In *International Conference on Robotics in Education (RiE).* Springer, 31–44.
[46] Anika Saxena, Chung Kwan Lo, Khe Foon Hew, and Gary Ka Wai Wong. 2020. Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher* 29, 1 (Feb. 2020), 55–66. https://doi.org/10.1007/s40299-019-00478-w
[47] Florian Scharf, Thomas Winkler, Claudia Hahn, Christian Wolters, and Michael Herczeg. 2012. Tangicons 3.0: An Educational Non-Competitive Collaborative Game. In *Proceedings of the 11th International Conference on Interaction Design and Children (IDC '12).* Association for Computing Machinery, New York, NY, USA, 144–151. https://doi.org/10.1145/2307096.2307113 event-place: Bremen, Germany.
[48] Florian Scharf, Thomas Winkler, and Michael Herczeg. 2008. Tangicons: Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class. In *Proceedings of the 7th International Conference on Interaction Design and Children (IDC '08).* Association for Computing Machinery, New York, NY, USA, 242–249. https://doi.org/10.1145/1463689.1463762 event-place: Chicago, Illinois.
[49] Edelberto Franco Silva, Bruno José Dembogurski, and Gustavo Silva Semaan. 2021. A Systematic Review of Computational Thinking in Early Ages. *arXiv:2106.10275 [cs]* (June 2021). http://arxiv.org/abs/2106.10275 arXiv: 2106.10275.
[50] Arnan Sipitakiat and Nusarin Nusen. 2012. Robo-Blocks: Designing Debugging Abilities in a Tangible Programming System for Early Primary School Children. In *Proceedings of the 11th International Conference on Interaction Design and Children (IDC '12).* Association for Computing Machinery, New York, NY, USA, 98–105. https://doi.org/10.1145/2307096.2307108 event-place: Bremen, Germany.
[51] Sphero. 2022. Screenless Programming & Learning Robot for Kids | Sphero indi. Retrieved April 29, 2022 from https://sphero.com/pages/sphero-indi
[52] Amanda Sullivan, Marina Bers, and Alex Pugnali. 2017. The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice* 16, 1 (2017), 171–193. Publisher: Informing Science Institute.
[53] Gonzalo Tejera, Guillermo Amorin, Andrés Sere, Nicolás Capricho, Pablo Margenat, and Jorge Visca. 2019. Robotito: programming robots from preschool to undergraduate school level. In *2019 19th International Conference on Advanced Robotics (ICAR).* IEEE, 296–301.
[54] TTS. 2022. Bee-Bot. Retrieved April 29, 2022 from https://www.tts-international.com/bee-bot-programmable-floor-robot/1015268.html?cgid=Primary-Computing_--_ICT-Bee-Bot_Blue-Bot_--_Pro-Bot
[55] TTS. 2022. Blue-Bot. Retrieved April 29, 2022 from https://www.tts-international.com/blue-bot-bluetooth-programmable-floor-robot/1015269.html
[56] TTS. 2022. Pro-Bot. Retrieved April 29, 2022 from https://www.tts-international.com/pro-bot-floor-robot-starter-pack/1010501.html?cgid=Primary-Computing_--_ICT-Bee-Bot_Blue-Bot_--_Pro-Bot
[57] VEX. 2022. 123 - VEX Robotics. Retrieved April 29, 2022 from https://www.vexrobotics.com/123
[58] Pia Williams, Sonja Sheridan, Heidi Harju-Luukkainen, and Ingrid Pramling Samuelsson. 2015. Does group size matter in preschool teacher's work? The skills teachers emphasise for children in preschool groups of different size. *Journal of Early Childhood Education Research* 4, 2 (2015), 93–108. Publisher: Suomen varhaiskasvatus.
[59] Jeanette Wing. 2011. Research notebook: Computational thinking—What and why. *The link magazine* 6 (2011), 20–23.
[60] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35. Publisher: ACM New York, NY, USA.
[61] Junnan Yu and Ricarose Roque. 2019. A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction* 21 (Sept. 2019), 17–36. https://doi.org/10.1016/j.ijcci.2019.04.001

Appendix 4

A Systematic Review of Technologies to Teach Control Structures in Preschool Education

Ewelina Bakala et al. «A Systematic Review of Technologies to Teach Control Structures in Preschool Education». In: *Frontiers in Psychology* 13 (2022), p. 911057

Author's contribution The author contributed to conception and design of the study. She was one of the two reviewers who analyzed the articles involved in the initial systematic literature review and evaluated the characteristics of the extracted tools. She also led and actively participated in the writing process.

frontiers | Frontiers in Psychology

SYSTEMATIC REVIEW
published: 16 September 2022
doi: 10.3389/fpsyg.2022.911057

A Systematic Review of Technologies to Teach Control Structures in Preschool Education

Ewelina Bakala[1,2]*, *Anaclara Gerosa*[2], *Juan Pablo Hourcade*[3], *Gonzalo Tejera*[1,2], *Kerry Peterman*[3] and *Guillermo Trinidad*[1]

[1] Facultad de Ingeniería, Instituto de Computación, Universidad de la República, Montevideo, Uruguay, [2] Centro Interdisciplinario en Cognición para la Enseñanza y el Aprendizaje, Universidad de la República, Montevideo, Uruguay, [3] Department of Computer Science, The University of Iowa, Iowa City, IA, United States

There is growing interest in teaching computational thinking (CT) to preschool children given evidence that they are able to understand and use CT concepts. One of the concepts that is central in CT definitions, is the concept of control structures, but it is not clear which tools and activities are successful in teaching it to young learners. This work aims at (1) providing a comprehensive overview of tools that enable preschool children to build programs that include control structures, and (2) analyzing empirical evidence of the usage of these tools to teach control structures to children between 3 and 6. It consists of three parts: systematic literature review (SLR) to identify tools to teach CT to young children, analysis of tools characteristics and the possibilities that they offer to express control structures, and SLR to identify empirical evidence of successful teaching of control structures to young children using relevant tools. This work provides an understanding of the current state of the art and identifies areas that require future exploration.

Keywords: control structures, young children, computational thinking, technology, systematic literature review, preschoolers

OPEN ACCESS

Edited by:
Stamatios Papadakis,
University of Crete, Greece

Reviewed by:
Cucuk Budiyanto,
Sebelas Maret University, Indonesia
Anastasios Theodoropoulos,
University of Peloponnese, Greece

***Correspondence:**
Ewelina Bakala
ebakala@fing.edu.uy

Specialty section:
This article was submitted to Educational Psychology, a section of the journal Frontiers in Psychology

Received: *01 April 2022*
Accepted: *06 May 2022*
Published: *16 September 2022*

Citation:
Bakala E, Gerosa A, Hourcade JP, Tejera G, Peterman K and Trinidad G (2022) A Systematic Review of Technologies to Teach Control Structures in Preschool Education. Front. Psychol. 13:911057. doi: 10.3389/fpsyg.2022.911057

1. INTRODUCTION

In 2006, Jeanette Wing popularized the term "Computational thinking" as a universal set of skills which could allow everyone to use computer science concepts for problem solving (Wing, 2006, 2011). Grover (2018) defined two viewpoints on CT: one is that CT is the cognitive or "thinking" counterpart to practicing computer science in CS classrooms. This means CT is a specific characteristic of practicing computer science and is bound to this discipline. The other viewpoint is that CT is a skill to be integrated by other disciplines and it is a way to approach sense-making in different subjects. Wing's original definition of CT was broad enough that it ignited educators and policy-makers' interest in CT (Bocconi et al., 2016). Thus, over the past decade there has been an increase in research around CT interventions targeted at most levels of formal education (Grover and Pea, 2013; Hsu et al., 2018; Yadav et al., 2018; Lyon and Magana, 2020; Stamatios, 2022), its inclusion within other disciplines (Orton et al., 2016; Weintrop et al., 2016; Hickmott et al., 2018), its association with other well-established cognitive skills (Román-González et al., 2017; Robertson et al., 2020; Gerosa et al., 2021; Tsarava et al., 2022), and focusing on creating reliable and valid assessment methods (Tang et al., 2020), amongst others. Moreover, both public and privately-led initiatives have been successfully implemented to foster CT in children and adolescents (Brackmann et al., 2016; Williamson, 2016), as it is regarded as a valuable twenty-first century skill (Yadav et al., 2016).

Several of the most widely accepted and cited definitions of CT propose the use and understanding of control structures such as loops and conditionals as an integral part of CT. For example, Brennan and Resnick (2012) named loops, conditionals and events as central computational concepts in their framework; Grover and Pea (2013) highlighted the use of conditional logic and iteration as well as Shute et al. (2017). In some cases there is no direct reference to control structures in CT definitions, but algorithm design (Khoo, 2020; Saxena et al., 2020) is considered as an essential part of CT. Control structures are basic building components for algorithms (Perkovic, 2015), and therefore an integral part of CT. Moreover, several of the assessments created for evaluating students' CT in formal education include the evaluation of loops and conditionals, such as Román-González (2015) and collaborators' CTt; Relkin et al.'s (2020) TechCheck or the CT sections that were incorporated to the PISA mathematics testing in OECD (2019).

Authors such as Bers (2019, 2020) have argued for the inclusion of CT skills in early childhood education, particularly through the use of robots as an embodied, tangible tool which would be intuitive and developmentally appropriate for young children. Teaching young children CT related concepts prepares them to solve real-life challenges in a logical and systematic way, and some authors consider CT as relevant as reading, writing and mathematics (Sanford and Naidu, 2016). The early exposure to computing has potential to engage both boys and girls mitigating gender-related barriers (Manches and Plowman, 2017; Martin et al., 2017).

This work aims at presenting the current state of the art of teaching control structures to preliterate children between 3 and 6 years of age using electronic tools (physical, virtual and hybrid systems) that allow users to construct explicit programs. Our work consists of three parts (see **Figure 1**): (1) review 1: a systematic literature review (SLR) of reviews aimed at identifying technology used to promote CT in young children; (2) technology overview: an analysis of the characteristics of these tools based on information we found in tool websites and user manuals; (3) review 2: a SLR of empirical evidence related to the use of the tools in teaching control structures to preliterate children between the ages of 3 and 6.

The research questions that guide this study are the following:

- What electronic tools exist to support the development of CT in young children? (review 1)
- Which tools are appropriate for preliterate children between the ages of 3 and 6? (technology overview)
- How can children introduce control structures into their programs using electronic tools? (technology overview)
- What tools have been reported to be successful for teaching control structures to young children? (review 2)

In the remainder of the paper we present related works that systematize the knowledge about existing tools that support

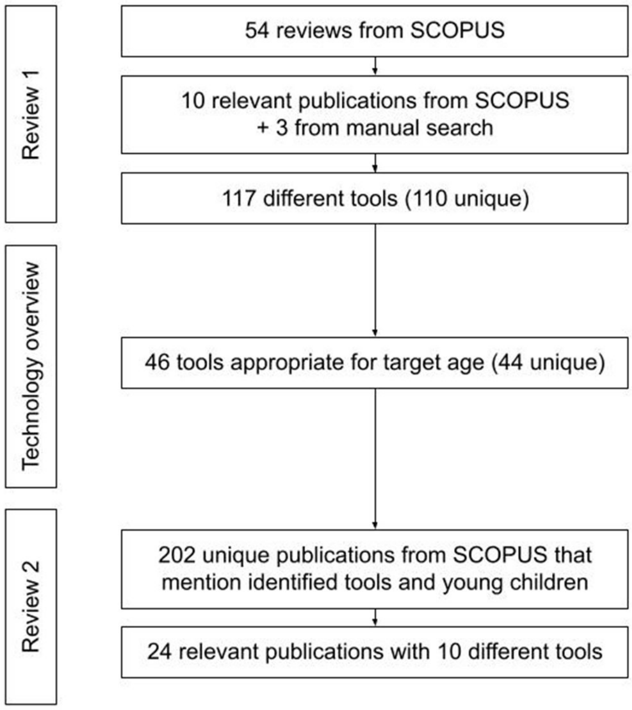


FIGURE 1 | An overview of the research pipeline.

the development of CT, next we present the methodology and findings of the first SLR that aims to identify existing tools for teaching CT to young children (see **Figure 1**). In the following step we analyze the tools to identify those that are electronic-based and appropriate for preliterate children between 3 and 6 years old, and provide details related to their price and possibilities that they offer to introduce control structures in children's code. The resulting list of appropriate tools is used in the second SLR to search for empirical evidence related to teaching control structures to young children. The limitations and results are discussed in the final section of the article and conclusions are laid down.

1.1. Related Work

Previous work has focused on reviewing technological and unplugged tools to promote CT in young children. However, most of the available reviews on this topic focus on the broad aspects of CT and robotics without specifically analyzing the affordances of particular technological tools for learning a specific concept, such as control structures. For example, Silva et al. (2021) focused on describing the available technology for 2–8 year old children as well as curricula implemented for these ages, while Kakavas and Ugolini (2019) focused on they way the teaching of CT has evolved in primary education in the last decades and was successful in identifying the context in which the technology was implemented and in which way CT was assessed. In a recent review (Bakala et al., 2021) we also analyzed the characteristics of robots and activities used in preschool education to promote CT skills with a focus on empirical research, however the specific ways in which each concept encompassed by CT was targeted was not part of our scope. Recent work by Taslibeyaz et al. (2020) shed light into the way studies with young children considered the concept of CT by analyzing its definitions, which skills were targeted and which variables were assessed and included the technological tools used to promote these skills. However, the implications as to how a specific technology causes this improvement and what are the nuances of using different technological tools were not discussed. Similarly, a recent review by Toh et al. (2016) on the use of robots for young children provided context on the type of study conducted and on the effects of robotics on children's cognitive outcomes as well as parents', educators' and children's opinions regarding the use of these tools. However, the possible benefits are discussed generally regarding robotics and this work does not focus on the outcomes of specific tools. Yu and Roque (2019) provide a comprehensive review of computational toys and kits for young children (7 and under) describing their design features, which computational concepts and practices they target and how they relate to other domains in knowledge. In particular, they analyzed the way conditionals were presented in the technological tools and argued that most of the time conditionals were implemented in an implicit way (thus not represented using explicit if-then statements). In addition, the authors explored the presentation of loops, pointing out many of the available tools used repeat blocks which encapsulated a given sequence, whether digital or concrete. In order to expand upon these findings, this review will focus specifically on the ways technology has implemented control structures and provide an overview of the evidence surrounding these implementations with young children. In this sense, our review will provide a summary of the empirical experiences in which these control structures have been taught to young children and analyze these results. To our knowledge, there isn't thus far a systematic review of literature which focuses on the implementation of control structures and provides a thorough analysis of how technological tools aimed at early childhood allow its users to learn them. In addition, we conducted a SLR on the existing empirical evidence in which control structures have been taught to children, shedding light into which practices and tools are supported by evidence and thus favorable for its inclusion in the classroom.

2. SLR OF EXISTING TOOLS (REVIEW 1)

We used a systematic literature review (Kitchenham et al., 2015) to answer our first research question: What tools exist to support the development of CT in young children?

2.1. Methodology

Systematic literature review (SLR) is a method that allows identifying relevant material to a given topic using an objective, analytical, and repeatable approach (Kitchenham et al., 2015). We carried out our literature review following the PRISMA guidelines (Moher et al., 2009). Four reviewers participated in the review process. Firstly, they defined the search term, inclusion and exclusion criteria, and data to extract from the publications. Secondly, two reviewers analyzed the publications to identify the relevant articles. One reviewer extracted the tools from relevant

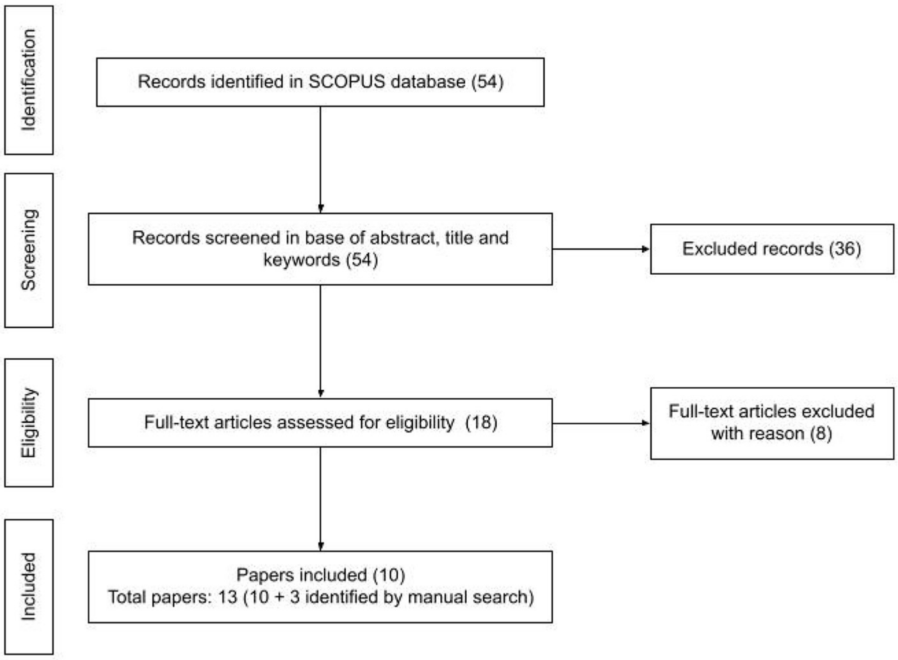


FIGURE 2 | Steps of the selection process of the first SLR. Reported in line with the PRISMA statement (Moher et al., 2009).

articles. A quality assessment stage was not included, as we were not interested in filtering out low quality studies since we still reviewed each tool or investigating changes in quality over time.

2.1.1. Search Strategy

To identify reviews of technology to support the development of CT in young children we applied an automated search (Kitchenham et al., 2015) in the Scopus search engine (Elsevier Scopus, 2022). The search term was the following:

TITLE-ABS-KEY (((review AND {computational thinking} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learners} OR {primary school} OR {primary education} OR k-6 OR k-8 OR childhood))))

We used three keywords: review, computational thinking and childhood (and synonyms) to search in the title, abstract, and keywords.

2.1.2. Study Selection

We defined the following inclusion criteria for the studies' selection:

- Articles that review electronic-based tools to promote the development of CT in young children.
- Publications focused on children between 3 and 5 years old, including 6 years old, if attending pre-primary school educational level.

Exclusion criteria were:

- Articles written in a language other than English or Spanish.
- Publications that target children older than 6 years.
- Articles limited to unplugged tools.
- Case studies.
- Conference proceedings.

The first round of the selection was made based on the information available in the abstract. Two researchers applied the criteria independently and filter out publications that do not review tools focused on promoting CT in young children. The articles were tagged as "relevant" or "irrelevant." If an article was classified differently by the reviewers, the full text was reviewed. If there were doubts about an article, they were discussed with two other reviewers that supervised this revision step. Also the articles that were considered relevant by both reviewers were analyzed in detail to confirm or reject their relevance.

2.1.3. Data Extraction

We used a spreadsheet to extract tools found in the publications and articles that mention each tool. We sorted each tool using categories that we developed (see Section 2.4).

2.2. Findings

2.3. Relevant Articles

The search was conducted on 6th of August 2021 and we obtained 54 articles to review (see **Figure 2**). In the screening phase the reviewers tagged identically 51 of 54 articles reaching an agreement of 0.94%. In the selection process we identified 10 articles relevant for this study. We added to our analysis 3 articles (Kakavas and Ugolini, 2019; Papadakis, 2020; Silva et al., 2021) that were identified by the manual search and that we considered a valuable source of information-Kakavas and Ugolini (2019) that was not indexed by Scopus, Papadakis (2020) that does not contain the word "review" in title, abstract and keywords and Silva et al. (2021) that is a preprint submitted to Elsevier.

A total of 13 articles (see **Table 1**) were used to elaborate the list of relevant tools. All the articles were published between 2018 and 2021.

2.4. Categories to Classify the Tools

To classify the tools we adapted and expanded categories proposed by Yu and Roque (2019). We obtained 4 main categories (see **Figure 3**): Physical, Virtual, Hybrid and No information. We divided Physical, Virtual and Hybrid into sub-categories and obtained 9 categories which we used to classify existing tools: Robots with tangible programming interface, Construction kits with no explicit program, Unplugged, Virtual with explicit program, Virtual with no explicit program, Robots with virtual programming interface, Construction kits with virtual programming interface, Virtual tools with tangible programming interface, No information. In the **Figure 3** there

TABLE 1 | 13 relevant publications that we identified in the first SLR.

References	Title
Bakala et al. (2021)	Preschool children, robots, and computational thinking: A systematic review
Papadakis (2021)	The Impact of Coding Apps to Support Young Children in Computational Thinking and Computational Fluency. A Literature Review
Fagerlund et al. (2021)	Computational thinking in programming with Scratch in primary schools: A systematic review
Yang et al. (2020)	The influence of robots on students' computational thinking: A literature review
Pedersen et al. (2020)	The effect of commercially available educational robotics: A systematic review
Umam et al. (2019)	Literature review of robotics learning devices to facilitate the development of computational thinking in early childhood
Isnaini et al. (2019)	Robotics-based learning to support computational thinking skills in early childhood
Yu and Roque (2019)	A review of computational toys and kits for young children
Ching et al. (2018)	Developing Computational Thinking with Educational Technologies for Young Learners
Ioannou and Makridou (2018)	Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work
Silva et al. (2021)	A Systematic Review of Computational Thinking in Early Ages
Papadakis (2020)	Robots and Robotics Kits for Early Childhood and First School Age
Kakavas and Ugolini (2019)	Computational thinking in primary education: a systematic literature review

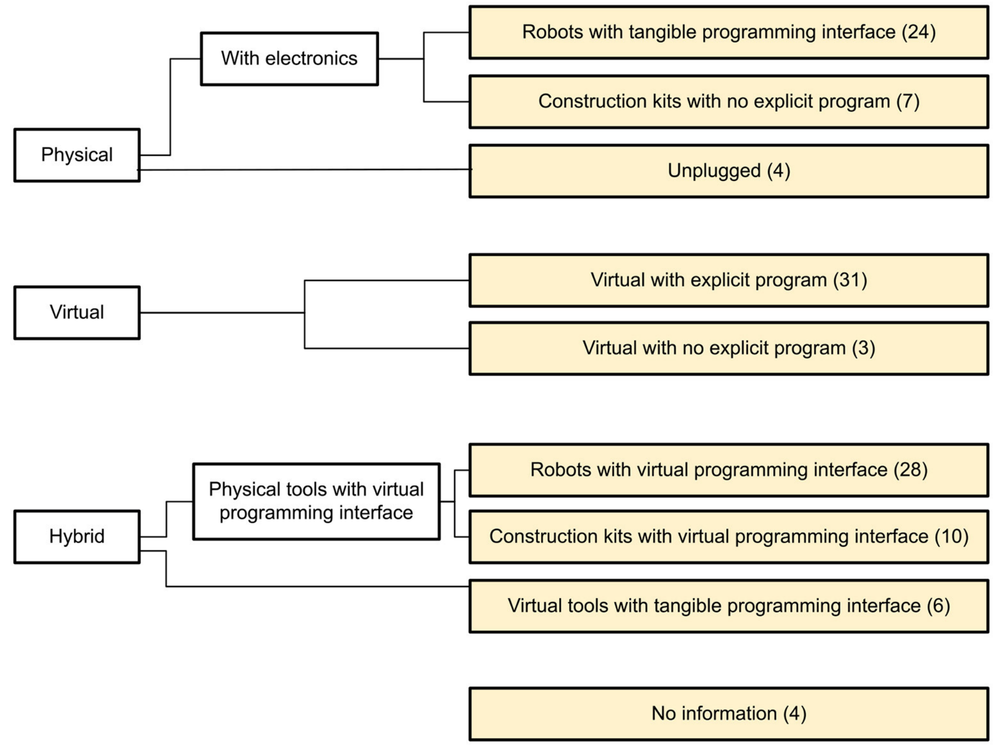


FIGURE 3 | Categories developed to classify the physical aspect of the tools.

are more than 8 categories, but only those highlighted in yellow were used to classify the tools.

We used the category Physical for tools that are fully tangible and do not require screen-based applications. We divided it into Unplugged and Physical tools with electronics. The last category was composed of Robots with tangible programming interface and Construction kits with no explicit program. The category Construction kits with no explicit program contains electronic building blocks that can be connected together to cause certain behavior of the system, but do not require the user to write an explicit program.

Virtual contains fully screen-based tools, such as desktop, mobile, or web apps. This category encompasses tools that do not require the user to write an explicit program (e.g., tools like CompThink App where the user has to solve logical problems without writing code) and those which need an explicit program.

Hybrid tools combine physical and virtual parts. We divided them into Virtual tools with tangible programming interface or Physical tools with virtual programming interface. The first category consists of applications with tangible programming interfaces. The second category is composed of Robots with virtual programming interface and Construction kits with virtual programming interface. The last category contains embedded systems like Arduino that can be programmed using a virtual programming interface.

The "No information" category was used if there was no information about the tool that could be used to classify it.

2.5. Tools

From the 13 relevant publications we extracted 110 tools (106 unique tools). In the case of Code & Go Robot Mouse, we found three different names that referred to this tool: Robot Mouse (Yu and Roque, 2019; Pedersen et al., 2020), Colby robotic mouse (Papadakis, 2020; Bakala et al., 2021) and Code & Go Robot Mouse (Ching et al., 2018; Silva et al., 2021), and we analyzed it as one single tool.

While reviewing the tools mentioned in the articles we found in external sources 4 more tools that we considered relevant for our work. We added Qobo (Physical and Hybrid), VEX 123 (Physical and Hybrid), Sphero indi (Physical and Hybrid), Scottie Go (Virtual) and ended up with a total of 117 tools (110 unique tools, see **Table 2**).

We classified 35 as Physical, 34 as Virtual, 44 as Hybrid and 4 as No information (see **Figure 3**).

It is important to say that seven tools were present in more than one category (Blue-Bot, Qobo, VEX 123, Sphero indi, VBOT, Makeblock Neuron, Tuk Tuk). For example, Blue-Bot is a robot that can be programmed using buttons on its back and because of that it belongs to the category Robots with tangible programming interface, but there is also a possibility to program it using an application, so it was also classified as a Robot with a virtual programming interface. That is why we refer to 110 unique tools, although we analyzed 117 relevant tools that included duplicated items.

In three cases (Ozobot, LEGO, Robotis and roboplus software) the names that we found in publications were names of brands, not names of specific tools, so it was impossible to classify them, and they were categorized as No information. One publication mentioned Robo Cup Junior as a tool. As far as we know RoboCup Junior (RoboCupJunior, 2022) is an educational initiative, not one particular technology, so we categorized this item as No information as well.

3. TECHNOLOGY OVERVIEW

The first aim of this part of our study was to identify how young, preliterate children can introduce conditionals and iterations into their programs using existing tools. This section is motivated by the following research questions:

- Which tools are appropriate for preliterate children between the ages of 3 and 6?
- How can children introduce control structures into their programs using electronic tools?

3.1. Methodology

Four reviewers participated in the revision of existing tools. Two of them reviewed the available online information and extracted the information of interest. The other two participated in the definition of the categories to classify tools' characteristics and helped to classify doubtful cases.

3.1.1. Tools Selection

We were interested in electronic tools that allow users to construct explicit programs, so we did not further analyze the tools classified as Unplugged, Construction kits with no explicit program, Virtual with no explicit program, and No information.

We identified the relevant tools by filtering out those not appropriate for children between 3 and 6 - tools that target children older than 6 years old or that should be programmed using interfaces that require reading skills (see **Table 2**). During tool selection we first analyzed the target age of each tool. If the information of the target age was expressed using educational levels like "elementary school" or "kindergarten" we translated this information into age using the United States educational system as reference. If the tool was designed for children older than 6, we tagged it as inappropriate and did not analyze its programming interfaces. If the age was of our interest, we proceeded with the inspection of the user interface. In many cases hybrid tools offered different programming languages/interfaces to cover a wide age spectrum of users, for example, Finch Robot can be programmed using 8 different programming languages and its promotional video states that it is suitable for users from "from kindergarten to college." In those cases we evaluated only programming languages appropriate for preliterate children. If there was no interface suitable for preschoolers, we marked it as a tool that requires reading skills.

3.1.2. Data Extraction

To collect the information about the tools we reviewed the official websites, video material provided by the manufacturer, online manuals, as well as, youtube videos and amazon websites.

During data extraction we were interested in classifying different types of control structures that can be used with each tool, so we defined categories that we present in Sections 3.2.2.1, 3.2.2.2.

3.2. Findings

3.2.1. Tools Selection

We identified 46 tools (44 unique) appropriate for preliterate children (see **Table 3**). Twenty Robots with tangible programming interface, 11 Virtual with explicit program and 15 Hybrid tools: 8 Robots with virtual programming interface, 1 Construction kit with virtual programming interface and 6 Virtual tools with tangible programming interface. Two tools (Blue Bot and Sphero indi) were classified as both: Robots with tangible programming interface and Robots with virtual programming interface.

There were three tools that we analyzed together: KIBO, KIWI and CHERP. KIBO is a robot currently available in the market, formerly known as "KIWI" or Kids Invent with Imagination (Tufts University, 2022). CHERP is a programming language that is used to program KIBO and KIWI, so evaluating CHERP is equivalent to evaluating KIBO and KIWI.

In the case of some tools, the programming interface contained images which made it accessible for preliterate children, but we had the impression that the systems were designed for children older than our target age. They contained text-based challenges (Scottie Go) and menus (BOTS, Neuron App, Move the turtle, RoboZZle), design that we consider unattractive for young children (RoboZZle, BOTS), text-based options with no associated image ("tap" event in Roberto), or comparisons involving high numeric values (Neuron App). Although these tools raised some doubts, we decided to include them in our analysis as we wanted to provide an inclusive overview of the existing tools.

3.2.2. Categories to Classify Control Structures

We developed categories related to the use of control structures to classify tools suitable for young children (see **Table 3**) that we identified during tools selection step (see Section 3.2.1).

3.2.2.1. Conditionals

To identify how the children can introduce decision making based on certain conditions into their programs we reviewed the programming interfaces and classified the existing tools with categories that we defined in an iterative process. Introducing conditions in the code was typically based on conditional branches (e.g., if-else structures) or based on events (e.g., blocking the program execution until some event occurs). From now on we will refer to those two forms of incorporation of conditions into the code as "conditionals."

To classify the degree of liberty that the children have while using and building conditionals in their programs, we propose three levels, ordered by increasing complexity for the user:

1. Predefined connection of condition and action: it is possible to use a predefined programming statement that connects an event with an action. For example, the Qobo robot detects coding cards below it and acts according to the statement stored in the card. It has a specific card for conditional turning

TABLE 2 | 117 tools extracted from 13 relevant publications that we identified in the first SLR.

Tool type	Name	Target age	Exclusion reason [Age, RRS (require reading skills), Unplugged, No info, No program]	Source
Robots with tangible programming interface	Bee Bot	3+		Umam et al. (2019), Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021), Bakala et al. (2021), Yang et al. (2020)
	Blue Bot	3–11		Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021)
	Botley	5+		Papadakis (2020)
	Code-a-Pillar	3–6		Ching et al. (2018), Yu and Roque (2019), Papadakis (2020)
	Cubetto	3–9		Isnaini et al. (2019), Ching et al. (2018), Yu and Roque (2019), Papadakis (2020), Umam et al. (2019), Pedersen et al. (2020)
	Dr. Wagon	6–12	RRS	Yu and Roque (2019)
	Edison robot	4–16	No program	Papadakis (2020), Pedersen et al. (2020)
	KIBO	4–7		Ching et al. (2018), Umam et al. (2019), Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021), Bakala et al. (2021), Yang et al. (2020);
	KIWI	5–7		Bakala et al. (2021)
	KUBO robot	4–10		Papadakis (2020), Pedersen et al. (2020)
	Matatalab Coding Set	4–9		Papadakis (2020)
	mTiny	4+		Papadakis (2020)
	Ozobot Evo	5–18		Papadakis (2020)
	Ozobot Bit	6+		Papadakis (2020), Bakala et al. (2021)
	Plobot	4+		Yu and Roque (2019)
	Pro-bot	3+		Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021)
	Qobo	3–8		Manual
	Roamer	4–13	No program	Papadakis (2020)
	Robot Mind Designer	7+	Age	Papadakis (2020)
	Code and Go Robot Mouse	4–9		Ching et al. (2018), Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021), Bakala et al. (2021);
	Robotito	4–6		Silva et al. (2021)
	Sphero indi	4–8		Manual
	TurtleBot	No info		Bakala et al. (2021)
	VEX 123	4–9		Manual
Construction kits with no explicit program	Cubelets	4+	No program	Papadakis (2020), Pedersen et al. (2020)
	Curlybot	No info	No program	Yu and Roque (2019)
	Electronic Blocks	4–6	No program	Yu and Roque (2019)
	LittleBits	8+	No program	Kakavas and Ugolini (2019), Pedersen et al. (2020)
	Makeblock Neuron	6+	No program	Pedersen et al. (2020)
	roBlocks	9+	No program	Yu and Roque (2019)
	Romibo	No info	No program	Pedersen et al. (2020)
Unplugged	Code Monkey Island	6+	Unplugged	Ching et al. (2018)
	Happy Maps	No info	Unplugged	Silva et al. (2021)
	Hello Ruby	5+	Unplugged	Yu and Roque (2019)
	Robot Turtles	4+	Unplugged	Ching et al. (2018), Yu and Roque (2019)
Virtual with explicit program	AgentCubes	8+	Age	Kakavas and Ugolini (2019)
	AgentSheets	11–13	Age	Kakavas and Ugolini (2019)
	Alice	11+	Age	Kakavas and Ugolini (2019)

(Continued)

TABLE 2 | Continued

Tool type	Name	Target Age	Exclusion reason [Age, RRS (require reading skills), Unplugged, No info, No program]	Source
	BOTS	5–18		Kakavas and Ugolini (2019)
	Cargo-Bot	10–18	Age	Ching et al. (2018), Yu and Roque (2019)
	Codeable Crafts	4+		Yu and Roque (2019)
	Code.org	4+		Ching et al. (2018), Silva et al. (2021)
	CodyColor	0+	No program	Silva et al. (2021)
	CTSiM	5–18	RRS	Kakavas and Ugolini (2019)
	Daisy the Dinosaur	7+	Age	Papadakis (2021)
	FormulaT Racing	7–13	Age	Kakavas and Ugolini (2019)
	Hopescotch	10–16	Age	Ching et al. (2018)
	Kodable	4–10		Ching et al. (2018), Papadakis (2021), Silva et al. (2021)
	Kodetu	9–17	Age	Kakavas and Ugolini (2019)
	Kodu	9+	Age	Kakavas and Ugolini (2019)
	Legato	4–11	No program	Ching et al. (2018), Silva et al. (2021)
	LightBot	9+	Age	Ching et al. (2018), Yu and Roque (2019), Kakavas and Ugolini (2019), Papadakis (2021), Silva et al. (2021)
	LightBotJr	4–8		Ching et al. (2018), Silva et al. (2021)
	MiniColon game	8–9	Age	Kakavas and Ugolini (2019)
	Move the turtle	5+		Yu and Roque (2019)
	RoboZZle	6–7		Yu and Roque (2019)
	Run Marco!	4+		Yu and Roque (2019)
	Scratch	8–16	Age	Ching et al. (2018), Isnaini et al. (2019), Kakavas and Ugolini (2019), Fagerlund et al. (2021)
	ScratchJr	5–7		Kakavas and Ugolini (2019), Yu and Roque (2019), Ching et al. (2018), Papadakis (2021), Silva et al. (2021)
	Story-Writing-Coding engine	5–11	RRS	Kakavas and Ugolini (2019)
	The Foos	5+		Yu and Roque (2019), Silva et al. (2021)
	Tuk Tuk (standard)	5–14	RRS	Silva et al. (2021)
	Tynker: Coding for Kids	5–14		Ching et al. (2018)
	VBOT	14+	Age	Ioannou and Makridou (2018), Yang et al. (2020)
	ViMAP	8–10	Age	Kakavas and Ugolini (2019)
	Zoombinis game	8+	Age	Kakavas and Ugolini (2019)
Virtual with no explicit program	CompThink App	5–11	No program	Kakavas and Ugolini (2019)
	PhysGramming	6–7	No program	Silva et al. (2021)
	Tuk Tuk (junior)	5–6	No prgram	Silva et al. (2021)
Robots with virtual programming interface	Blue Bot	3–11		Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020), Silva et al. (2021)
	CHERP	5–6		Ioannou and Makridou (2018), Kakavas and Ugolini (2019)
	Codey Rocky	5–11	RRS	Pedersen et al. (2020)
	COJI	6+		Yu and Roque (2019), Papadakis (2020)
	Cozmo	8–11	Age	Pedersen et al. (2020)
	Dash and/or Dot	6+		Ching et al. (2018), Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020)
	Finch	5+		Papadakis (2020)
	LEGO Boost	7–12	Age	Pedersen et al. (2020)
	LEGO Education WeDo	7+	Age	Kakavas and Ugolini (2019), Isnaini et al. (2019), Ching et al. (2018), Papadakis (2020), Silva et al. (2021), Pedersen et al. (2020), Umam et al. (2019), Bakala et al. (2021)

(Continued)

TABLE 2 | Continued

Tool type	Name	Target Age	Exclusion reason [Age, RRS (require reading skills), Unplugged, No info, No program]	Source
	LEGO Mindstorm	10+	Age	Kakavas and Ugolini (2019), Ching et al. (2018), Ioannou and Makridou (2018), Pedersen et al. (2020), Bakala et al. (2021)
	Max Tobo coding robot	6+	RRS	Papadakis (2020)
	mBot	8+	Age	Pedersen et al. (2020), Silva et al. (2021)
	MeeperBots	5–12	RRS	Yu and Roque (2019), Papadakis (2020)
	Mind designer robot	7+	Age	Papadakis (2020)
	MiP	8–15	Age	Pedersen et al. (2020)
	MU Spacebot	8+	Age	Pedersen et al. (2020)
	NAO	5–18	RRS	Kakavas and Ugolini (2019), Pedersen et al. (2020)
	Qobo	3–8	RRS	Manual
	ROBOTC Graphical	No info	RRS	Kakavas and Ugolini (2019)
	Scribbler	14+	Age	Pedersen et al. (2020)
	Sphero Ollie	8–14	Age	Pedersen et al. (2020)
	Sphero indi	4–8		Manual
	Sphero mini	8+	Age	Papadakis (2020)
	The Coffee Platform	No info	RRS	Ioannou and Makridou (2018)
	Thymio	6+		Yu and Roque (2019), Papadakis (2020), Pedersen et al. (2020)
	Tinkerbots	5+		Papadakis (2020)
	VEX 123	4–9	RRS	Manual
	VBOT	11–18	Age	Ioannou and Makridou (2018), Yang et al. (2020)
Construction kits with virtual programming interface	An ultra-low cost line follower Robotic	16–18	Age	Yang et al. (2020)
	Arduino+scratch	7–13	Age	Yang et al. (2020)
	CyberPLAYce	8–12	Age	Kakavas and Ugolini (2019)
	GoGo Board	10–18	Age	Ioannou and Makridou (2018)
	Hummingbird Robotics Kit	9–18	RRS	Pedersen et al. (2020)
	Makeblock Neuron	6+		Pedersen et al. (2020)
	micro:bit	8–14	Age	Pedersen et al. (2020)
	Scratch 4 Arduino, S4A)	8–17	Age	Kakavas and Ugolini (2019)
	ultimate	12+	Age	Pedersen et al. (2020)
	VEX IQ	11+	Age	Pedersen et al. (2020)
Virtual tools with tangible programming interface	Puzzlets Starter Pack	6+		Yu and Roque (2019)
	Roberto	4+		Yu and Roque (2019)
	Scottie Go	4–15		Manual
	Coding Awbie	5–11		Ching et al. (2018), Papadakis (2020), Silva et al. (2021), Yu and Roque (2019)
	Tabletop puzzle block system	4–5		Yu and Roque (2019)
	T-Maze	5–9		Kakavas and Ugolini (2019), Silva et al. (2021)
No info	LEGO	No info	No info	Yang et al. (2020), Bakala et al. (2021)
	Ozobot	No info	No info	Kakavas and Ugolini (2019), Pedersen et al. (2020)
	Robo Cup Junior	no info	No info	Isnaini et al. (2019)
	Robotis and roboplus software	No info	No info	Ioannou and Makridou (2018)

TABLE 3 | An overview of 46 relevant tools considering their price and possibilities to incorporate control structures into the code.

Classification	Tool name	Conditionals [Predefined connection, Free connection, Free condition building]	Integration with the main program [Integrated if, Blocking event, Interruption, Parallel execution]	Number of repetitions [Fixed number of repetitions, Configurable number of repetitions, Infinite loop]	Number of repeated commands [Single command repetition, Multiple command repetition]	Price (USD)
Robots with tangible programming interface	Bee Bot	–	–	–	–	85
	Blue Bot	–	–	–	–	104
	Botley	Free connection	Interruption	Configurable number of repetitions	Multiple command repetition	47
	Code–a–Pillar	–	–	Configurable number of repetitions	Single command repetition	148 (new version) or 35 (old)
	Cubetto	–	–		–	225
	KIBO	Free connection + Free condition building	Blocking event + Integrated if	Configurable number of repetitions + Infinite loop	Multiple command repetition	230 to 610
	KIWI	Free connection + Free condition building	Blocking event + Integrated if	Configurable number of repetitions + Infinite loop	Multiple command repetition	Unavailable
	KUBO robot	–	–	Configurable number of repetitions	Multiple command repetition	310 to 396
	Matatalab Coding Set	Free connection + Free condition building	Blocking event	Configurable number of repetitions	Multiple command repetition	169
	mTiny	–	–	Configurable number of repetitions	Multiple command repetition	120
	Ozobot Evo	Predefined connection	Integrated if	–	–	175
	Ozobot Bit	Predefined connection	Integrated if	–	–	Unavailable
	Plobot	Free connection	Blocking event	–	–	Unavailable
	Pro–bot	Free connection	Interruption	Configurable number of repetitions	Multiple command repetition	150
	Qobo	Predefined connection	Blocking event + Integrated if	Fixed number of repetitions	Multiple command repetition	60
	Robot Mouse	–	–	–	–	60
	Robotito	Predefined connection	Interruption	–	–	Unavailable
	Sphero indi	Predefined connection	Interruption	–	–	100
	TurtleBot	Predefined connection	Integrated if	–	–	105
	VEX 123	Free connection	Integrated if	Fixed number of repetitions + Configurable number of repetitions + Infinite loop	Single command repetition + Multiple command repetition	119
Virtual with explicit program	BOTS	Free condition building	Integrated if	Configurable number of repetitions	Multiple command repetition	Unavailable
	Codeable Crafts	Free connection	Parallel execution	Configurable number of repetitions + Infinite loop	Single command repetition + Multiple command repetition	Free
	Code.org	Free condition building	Interruption	Configurable number of repetitions	Multiple command repetition	Free
	Kodable	Free connection	Interruption	Configurable number of repetitions	Multiple command repetition	Free–2000 yearly
	LightBotJr	–	–	Configurable number of repetitions + Infinite loop	Multiple command repetition	2.99
	Move the turtle	Free condition building	Integrated if	Configurable number of repetitions	Multiple command repetition	3.99
	RoboZZle	Free connection	Interruption	Configurable number of repetitions + Infinite loop	Multiple command repetition	Free
	Run Marco!	Free condition building	Integrated if	Configurable number of repetitions	Multiple command repetition	Free

(Continued)

TABLE 3 | Continued

Classification	Tool name	Conditionals [Predefined connection, Free connection, Free condition building]	Integration with the main program [Integrated if, Blocking event, Interruption, Parallel execution]	Number of repetitions [Fixed number of repetitions, Configurable number of repetitions, Infinite loop]	Number of repeated commands [Single command repetition, Multiple command repetition]	Price (USD)
	ScratchJr	Free connection	Parallel execution	Configurable number of repetitions + Infinite loop	Single command repetition + Multiple command repetition	Free
	The Foos	Free condition building	Integrated if	Configurable number of repetitions + Infinite loop	Multiple command repetition	Free
	Tynker: Coding for Kids	Free connection	Integrated if + Interruption	Configurable number of repetitions	Single command repetition + Multiple command repetition	Free
Robots with virtual programming interface	Blue Bot	–	–	Configurable number of repetitions	Multiple command repetition	104
	CHERP	Free connection + Free condition building	Blocking event + Integrated if	Configurable number of repetitions + Infinite loop	Multiple command repetition	Unavailable
	COJI	Free connection	Interruption	–	–	32
	Dash and/or Dot	Free connection + Free condition building	Blocking event	Infinite loop	Multiple command repetition	150
	Finch	Free connection	Parallel execution	Configurable number of repetitions	Multiple command repetition	139
	Sphero indi	Free connection	Interruption	–	–	100
	Thymio	Free connection	Interruption	–	–	160
	Tinkerbots	–	–	Configurable number of repetitions	Single command repetition + Multiple command repetition	149
Construction kits with virtual programming interface	Makeblock Neuron	Free condition building	Integrated if	–	–	Unavailable
Virtual tools with tangible programming interface	Puzzlets Starter Pack	–	–	Configurable number of repetitions	Single command repetition	147
	Roberto	Free condition building	Blocking event	Infinite loop	Multiple command repetition	Unavailable
	Scottie Go	Free condition building	Integrated if	Configurable number of repetitions + Infinite loop	Single command repetition + Multiple command repetition	45–74
	Coding Awbie	Free connection	Integrated if	Configurable number of repetitions	Single command repetition + Multiple command repetition	99
	Tabletop puzzle block system	–	–	–	–	Unavailable
	T–Maze	Predefined connection	Blocking event	Configurable number of repetitions	Multiple command repetition	Unavailable

- if the robot passes over a card with a banana before passing over a bifurcation card, it turns left, but if it passes over a card with an apple, it turns right. Neither the condition nor the resulting action can be modified by the user.

2. Free connection of predefined condition and predefined action: it is possible to combine predefined conditions with predefined actions to build custom conditionals. For example, the Sphero Edu Jr application (see **Table 5**) allows users to associate a color sensed by the robot (predefined condition) with an action involving movement, light, and/or sound of the Sphero indi robot (predefined actions). The user needs at least two programming statements (condition and action)

to build a conditional. In the case of Kodable and RoboZZle these two statements are combined in one coding block: the background color of the block defines the condition (e.g., "if the tile is pink") and the arrow, the action (e.g., "go right"). The user is able to modify both: the background color and the arrow direction (see **Table 5**).

3. Free condition building: there are blocks that have to be combined with condition and action. In these cases the user has to use at least three components (bridge-block, condition, and action) to define a conditional. For example, the Matatalab Coding Set contains a "wait until" block that should be combined with a condition (e.g., dark or light) and a sequence of actions in order to build conditionals.

We provide the description and graphical example for each tool that supports conditionals in three tables: **Table 4** gathers tools that implement the first level, **Table 5** corresponding to the second level, and **Table 6** corresponding to the last one.

The only tools that enable the definition of conditionals using logical operators (e.g., AND, OR) were Makeblock Neuron and Thymio. Neuron online mode allows users to program behaviors using Neuron App, which supports multiple conditions. In the case of Thymio, the user has to associate events sensed by the robot with its behavior. It is possible to combine the sensing and internal state of the robot (e.g., if Thymio touched AND internal state equal to 1) to program advanced robot responses.

In the case of BOTS, Move the turtle, and Makeblock Neuron + Neuron (app) conditionals are based on numerical variables (e.g., $a > 5$) which makes them more complex than conditionals with non-numerical conditions (e.g., "if the sensed color is red"), as the children have to understand the concept of variable.

In the case of Coding Awbie, the Caution Block is the only means to introduce conditionals into the code, and is a phased out feature as the block is not included in new kits (Getting Started with Osmo Coding Awbie Manual, 2022).

We also analyzed how the code related to a certain condition interacts with the main program, and identified that they occur within either event-based or procedural programming paradigms. Within event-based programming, we identified the following categories:

- Blocking event: the main program contains a condition that blocks the execution until the condition is fulfilled. For example, KIBO contains a "wait for clap" block that makes the robot wait for a clap before executing commands stored in the following blocks.
- Interruption: the main program is interrupted when a certain event occurs. For example, in the case of Pro-bot the main program is interrupted if the sound sensor is triggered and the procedure associated with this event is executed.
- Parallel execution: It is possible for an event to lead to actions to occur in parallel or in addition to those already occurring. For example, an event in Scratch Jr. could generate a sound while a sprite continues moving on the screen.

Using a procedural programming paradigm, we identified the following category:

- Integrated if: the main program contains conditions expressed using the "if" structure that is evaluated during the program's execution. For example, KIBO allows to incorporate an if-statement into the sequence of commands. If the condition that is evaluated is true, the conditional code is executed and then, the remaining statements.

3.2.2.2. Loops

Another control structure that was relevant for us to analyze was the availability of loops enabling the iteration of commands.

We observed two modalities of implementing the iteration of commands:

- Single command repetition: the tool does not provide the possibility to repeat a sequence of commands, it allows only the repetition of a single action.
- Multiple command repetition: it is possible to repeat multiple commands. In this category we find tools that, due to the design of loop structure, limit the number of pieces that can be repeated (e.g., in Kodable the user is allowed to repeat only two commands) and tools that do not have this restriction.

We also analyzed how the amount of repetitions can be expressed:

- Fixed number of repetitions: the number of repetitions is fixed and cannot be changed by the user.
- Configurable number of repetitions: the amount of repetitions can be defined by the user.
- Infinite loop: it is possible to build infinite loops.

We provide an example for each category in the **Table 7**.

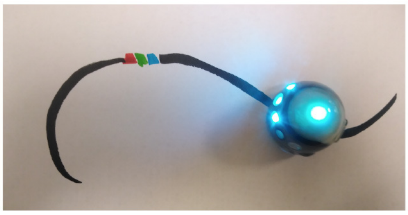
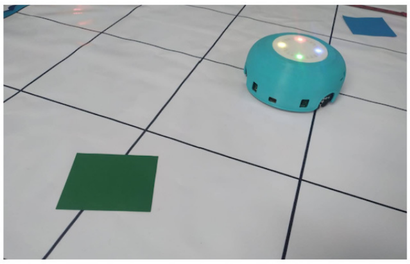
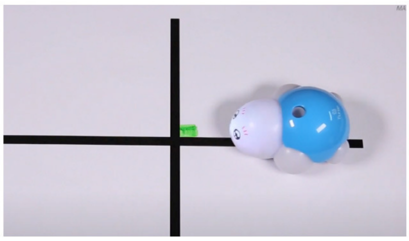
In most cases the amount of repetitions was expressed by associating the number of repetitions with a sequence of statements (similar to a for loop in more advanced programming languages), only BOTS uses exclusively conditions to stop the iteration process (similar to a while loop). KIBO, Finch, Run Marco!, Tynker: Coding for Kids, Scottie Go and VEX 123 offer both types ("repeat X times" and "repeat while") of repetition statements.

We found many different ways to implement infinite loops: using repeat forever (ScratchJr) or "go to start" command (VEX 123) at the end of the program, elements that contain pieces of code equivalent to "repeat forever" command (Roberto, Code.org), by building circular transitions between states (Dash and Dot), or by calling auxiliary functions (LightbotJr, RoboZZle).

3.3. Cost and Availability

Some tools that we analyzed are currently not available for sale: Plobot is a Kickstarter project that finished in Kickstarter (2022), Robotito, BOTS, Roberto, and T-Maze are academic developments, KIWI is KIBO's predecessor and is no longer manufactured, Makeblock Neuron and Puzzlets Starter Pack do not appear in online stores and CHERP is a programming language for KIBO and is not sold separately. All these tools were tagged as "unavailable."

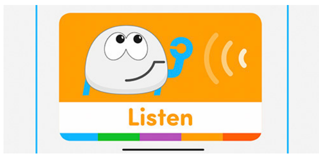
TABLE 4 | Tools that allow building conditionals categorized as "Predefined connection of condition and action".

Tool	Description	Reference image
Qobo	Specific card for conditional turning - if the robot passes over a card with a banana before passing over a bifurcation card, it turns left, but if it passes over a card with an apple, it turns right.	
Sphero indi	Color cards that the robot senses in the environment code robots' actions. Image provided by Sphero (2022).	
Ozobot Bit and Evo	Color lines that the robot senses in the environment code robots' actions.	
Robotito	Color cards that the robot senses in the environment code robots' actions.	
TurtleBot	Color codes that the robot senses in the environment code robots' actions.	

4. SLR OF EMPIRICAL EVIDENCE (REVIEW 2)

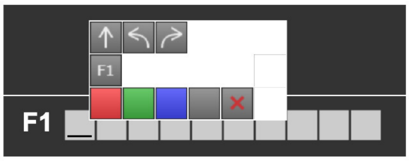
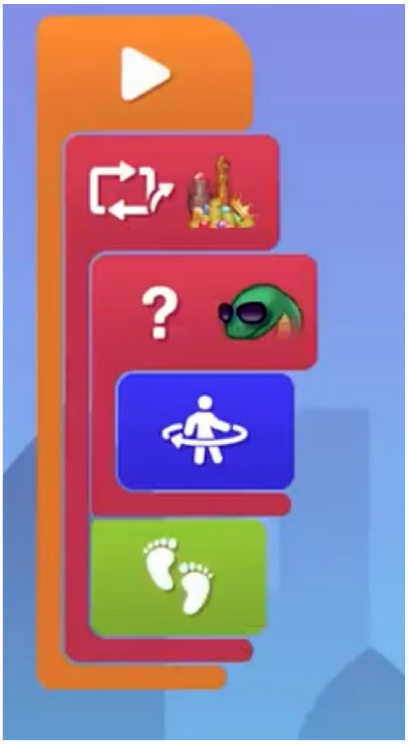
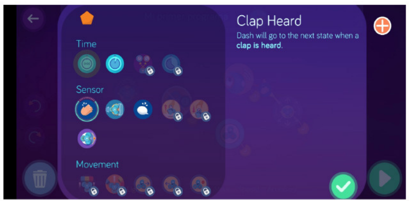
We conducted a second SLR (see **Figure 4**) to identify literature that reports empirical studies with tools that we considered relevant (see **Table 3**), in which control structures were taught and/or evaluated in order to respond the following research question: What tools have been reported to be successful for teaching control structures?

TABLE 5 | Tools that allow building conditionals categorized as “Free connection of predefined condition and predefined action”.

Tool	Description	Reference image
KIBO	“Wait for clap” block stops the program execution until the clap is sensed.	
Botley	Botley’s control provides an “object detection” button that is used to store the program that is executed when an obstacle is detected in front of the robot.	
Matatalab Coding Set	Two robots can send messages to each other. “Message received” block is used to define the robot’s action when a message is received. The block is available in Matatalab Sensor Add-on (2022).	
Plobot	“Listen” card blocks the program execution until Plobot detects a sound louder than a soft clap.	
Pro-bot	Procedure numbers 33 to 37 are activated with sensors. For example, the procedure associated with a light sensor runs when the light sensor goes from dark to light.	
VEX 123	Control cards make use of sensors to check conditions.	
ScratchJr and Codeable Crafts	Events related to characters like “on bump” or “on tap” can be associated with actions.	

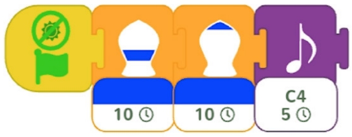
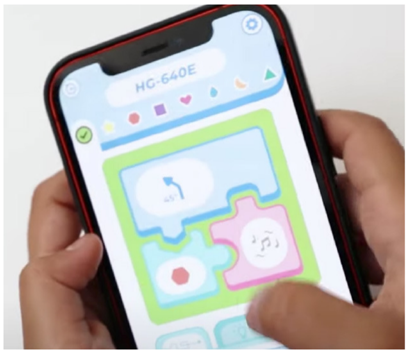
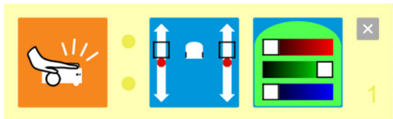
(Continued)

TABLE 5 | Continued

Tool	Description	Reference image
Kodable	The background color of the block defines the condition (e.g., "if the tile is pink") and the arrow, the action (e.g., "go right"). Image used with permission of Kodable (2022).	
RoboZZle	The background color of the block defines the condition (e.g., "if the tile is red") and the arrow, the action (e.g., "turn right").	
Tynker: Coding for Kids	Predefined condition (e.g., "if snake") can be combined with an action.	
COJI + COJI robot app	Predefined events can be associated with actions, for example, if the head is touched (event that activates procedure 1) - turn and sing (actions defined by the user).	No authorization for the image use.
Dash and Dot + Wonder for Dash & Dot Robots	Robot's actions are defined as states and the transition between can be fired based on conditions like "clap heard."	

(Continued)

TABLE 5 | Continued

Tool	Description	Reference image
Finch + Finchblox	Blocks attached to the "start when dark" block will be executed when the Finch detects that it is dark.	
Sphero indi + Sphero Edu Jr	Sphero Edu Jr application allows users to associate a color sensed by the robot with an action involving movement, light, and/or sound.	
Thymio + Thymio VPL	The user can associate events with actions.	
Coding Awbie	Caution Block enables a choice between two sets of sequences based on if there's an obstacle. Image can be found in Getting Started with Osmo Coding Awbie Manual (2022).	No authorization for the image use.
T-maze	"In a program execution, when the avatar reaches one of these squares in the maze, the child must do something with the sensors (e.g., cover a light sensor) to allow the avatar to proceed" Wang et al. (2014).	

4.1. Methodology

Two reviewers reviewed abstracts and tagged them as "irrelevant" or "relevant." The last category was used in the cases of publications that meet inclusion criteria or when it was impossible to evaluate the article relevance based on the information available in the abstract. One reviewer reviewed studies that were classified differently among reviewers in the previous step and tried to resolve the doubtful cases. If it was impossible, the articles were considered as "relevant" cases. One reviewer reviewed full-texts of relevant publications and took the final decision about their relevance for this study. We decided not to carry out any quality assessment of the studies as we wanted to provide a broad view of the existing empirical evidence. Two reviewers extracted the data.

4.1.1. Search Strategy

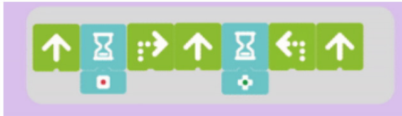
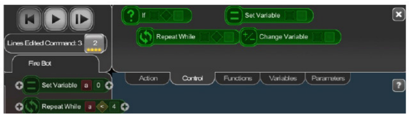
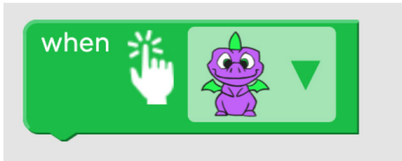
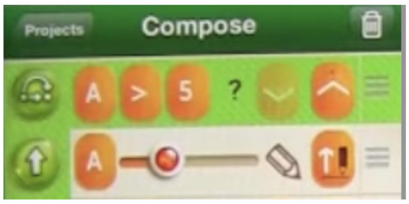
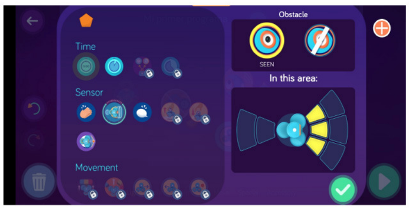
We used an automated search (Kitchenham et al., 2015) in Scopus search engine (Elsevier Scopus, 2022) to identify empirical studies with preschoolers that were developed using tools that we considered relevant (see **Table 3**). The search term was the following:

TITLE-ABS-KEY ((({**Tool name**} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))

It had two keywords: tool name and young learners (and synonyms) and was used to search in title, abstract and keywords.

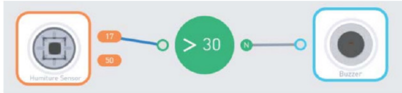
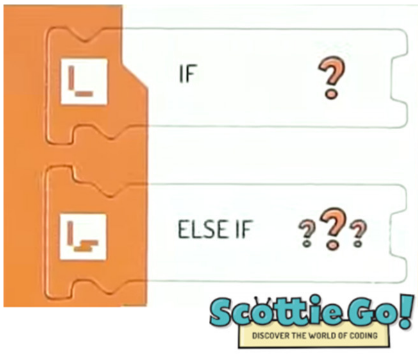
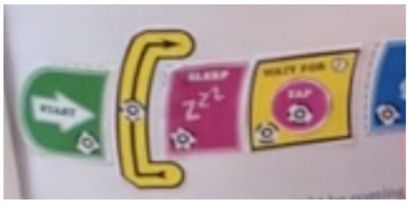
In some cases we used curly brackets, that limit the search to exact words, ignoring spelling variation or plurals, around the name of the tool ({Tool name}) to avoid false positive results. For example, in the case of "Coffee Platform" when we used Coffee AND Platform instead of Coffee Platform, the results contained irrelevant publications that did not target the robotic platform. In some cases we excluded publications from areas related to medicine, as some tools' names were equal to terms used in medicine and also brought false positive results (as in the case of

TABLE 6 | Tools that allow building conditionals categorized as "Free condition building".

Tool	Description	Reference image
KIBO	"If" block provides place to add a condition (e.g., far, near, dark, light).	
Matatalab Coding Set	"Wait until" can be connected with conditions like: dark, light, obstacle, etc. The block is available in Matatalab Sensor Add-on (2022).	
BOTS	"If" block should be associated with variable comparison (e.g., a > 5).	
Code.org	The condition in "when tapped" can be modified.	
Move the turtle	Condition block evaluates the value of a variable (A > 5).	
Run Marco!	"If" block can be modified.	
The Foos	The condition is variable and can be changed by the user. A video reference of the implementation can be found on CodeSpark Academy Youtube Channel (2022).	No authorization for the image use.
Dash and Dot + Wonder for Dash & Dot Robots	Robot's actions are defined as states and the transition between can be fired based on conditions like "obstacle detection" that can be customized (obstacle seen vs no obstacle, obstacle seen close vs far).	

(Continued)

TABLE 6 | Continued

Tool	Description	Reference image
Makeblock Neuron + Neuron (app)	Users can define conditions to establish relations between sensors and actuators.	
Scottie Go	"If" block should be associated with a specific condition.	
Roberto	"Wait for" can be combined with "tap" event.	

T-Maze). The search term used and the amount of publications found with each tool can be consulted in **appendix**.

4.1.2. Study Selection

The inclusion criteria for the studies' selection were the following:

- Articles that report empirical studies with young children using an electronic-based tool that enables activities with control structures.
- Publications that report activities or evaluations focused on control structures.
- Publications focused on children between 3 and 5 years old, including 6 years old, if attending pre-primary school educational level.

Exclusion criteria were:

- Publications that target children older than 6 years.
- Publications that do not report activities or evaluations focused on control structures.
- Off topic articles.
- Articles that describe experiences with users with neurodevelopmental disorders.
- Articles written in a language other than English or Spanish.
- Conference proceedings.

4.1.3. Data Extraction

In the data extraction step we used a spreadsheet to collect information related to the age of participants, number of participants, type of the study, learning outcome, activities aimed at programming conditions, activities that incorporate iterations. Based on the extracted data, two researchers conducted a thematic analysis to summarize study results.

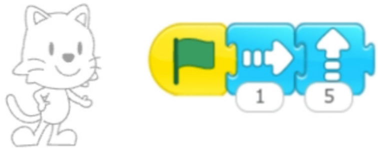
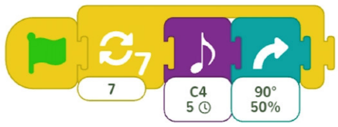
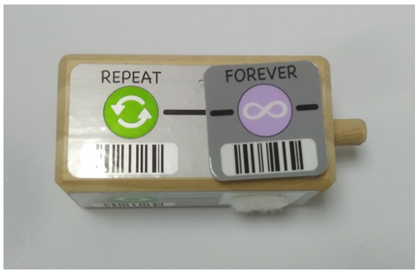
4.2. Findings

4.2.1. Scopus Search Result

The Scopus search for all tools was conducted on 13th of October 2021. In many cases the search brought no results. Only 26 tools of 44 unique tools that we identified, counted with Scopus entries (see **Appendix**). A total of 205 (202 unique) publications were analyzed. Three publications appeared as repeated because the research that they described involved two relevant tools, for example, Pugnali et al.'s research involved KIBO and ScratchJr, so it was found under the search query for KIBO and ScratchJr. We identified 24 unique publications (see **Table 8**) that met all inclusion criteria. In the screening phase the reviewers identically tagged 152 of 202 unique articles reaching an agreement of 75%.

The 24 relevant publications reported experiences with 10 different tools that we identified as relevant: ScratchJr (evaluated in 7 articles), KIBO (8), KIWI (2), CHERP (2), Code.org (2), BOTS (1), Kodable (1), Move the turtle (1), Strawbies (1) and T-maze (1). Strawbies is an alternative name for Coding Awbie that was used for the search, as the search term with "Coding Awbie" brought no results. Daisy the Dinosaur was mentioned in a study related to Kodable (Pila et al., 2019), but it targets older children (see **Table 2**). We also found one case of a custom tool

TABLE 7 | Examples of tools for categories developed to classify code iteration.

Category	Description	Reference image
Single command repetition	ScratchJr direction blocks can be modified to make more than one step using single block.	
Multiple command repetition	Kodable allows to repeat two commands.	
Fixed number of repetitions	Qobo coding card with fixed number of repetitions.	
Configurable number of repetitions	Finchblox allows to modify the number of repetitions.	
Infinite loop	KIBO allows to associate the repeat block with an infinity symbol.	

(Rose et al., 2017): a game with both ScratchJr-like and Lightbot style programming interface.

4.2.2. Thematic Analysis

4.2.2.1. KIBO/CHERP/KIWI Articles

The only set of technologies for which control structures have been evaluated multiple times was KIBO/CHERP/KIWI, developed by Marina Bers' group at Tufts University. Of the articles we identified using this technology, five evaluated children's use of control structures while separating the performance of young children from that of older children, or only including children within our inclusion criteria. All these evaluations used the Solve-It assessments, which were developed by the same research group. Through these assessments, in four of the publications, children who fit our inclusion criteria demonstrated proficiency when programming repeat loops (with a given number of repetitions) and "wait for clap" programs, but were not tested on or were unable to be proficient in the use of sensor loops or conditionals (Strawhacker and Bers, 2015; Elkin et al., 2016; Sullivan and Bers, 2016b; Bers et al., 2019). There was one outlying study where children in Kindergarten were able to demonstrate proficiency across all Solve It assessment areas, including repeat loops, sensor loops, "wait for clap" programs, and conditionals (Sullivan and Bers, 2018). Four other evaluations of this tool did not include specific evaluations of control flow (Kazakoff and Bers, 2014; Sullivan et al., 2017; Bers, 2019; Jurado et al., 2020) while two others did not separate children in our age range of interest from older children.

TABLE 8 | 24 relevant publications that we identified in the second SLR.

References	Title	Tool name	Type of tool	Age of participants	Number of participants
Jurado et al. (2020)	Social steam learning at an early age with robotic platforms: A case study in four schools in Spain	KIBO	Physical	4–6	65
Bers (2019)	Coding as another language: a pedagogical approach for teaching computer science in early childhood	KIBO, Scratch Jr	Physical, virtual	4–7	at least 9
Sullivan and Bers (2019)	Investigating the use of robotics to increase girls' interest in engineering during early elementary school	KIBO	Physical	5–7	105
Bers et al. (2019)	Coding as a playground: Promoting positive learning experiences in childhood classrooms	KIBO	Physical	3–5	172
Sullivan and Bers (2018)	Dancing robots: integrating art, music, and robotics in Singapore's early childhood centers	KIBO	Physical	3–6	98
Sullivan et al. (2017)	Imagining, playing, and coding with kibo: Using robotics to foster computational thinking in young children	KIBO	Physical	3–7	322
Pugnali et al. (2017)	THE impact of user interface on young children's computational thinking	KIBO, Scratch Jr	Physical, virtual	4–7	28
Elkin et al. (2016)	Programming with the KIBO Robotics Kit in Preschool Classrooms	KIBO	Physical	3–5	64
Sullivan and Bers (2016b)	Robotics in the early childhood classroom: learning outcomes from an 8–week robotics curriculum in pre–kindergarten through second grade	KIWI	Physical	4–7	60
Sullivan and Bers (2016a)	Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks	KIWI, BOTS	Physical, virtual	4–7	45
Strawhacker and Bers (2015)	"I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces	CHERP	Hybrid	5–6	35
Kazakoff and Bers (2014)	Put your robot in, put your robot out: Sequencing through programming robots in early childhood	CHERP	Hybrid	4–6	34
Arfé et al. (2020)	The effects of coding on children's planning and inhibition skills	Code.org	Virtual	5–6	179
Çiftci and Bildiren (2020)	The effect of coding courses on the cognitive abilities and problem–solving skills of preschool children	Code.org	Virtual	4–5	28
Pila et al. (2019)	Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children	Kodable, Daisy the Dinosaur	Virtual	4–6	28
Jung et al. (2019)	TurtleTalk: An educational programming game for children with voice user interface	Move the turtle	Virtual	6–9	8
Strawhacker and Bers (2019)	What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children	ScratchJr	Virtual	5–8	57
Pinto and Osório (2019)	Learn to program in preschool: Analysis with the participation scale [Aprender a programar en educación infantil: Análisis con la escala de participación]	ScratchJr	Virtual	3–6	71
Strawhacker et al. (2018)	Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr	ScratchJr	Virtual	5–7	222
Rose et al. (2017)	An exploration of the role of visual programming tools in the development of young children's computational thinking	Game with ScratchJr– and Lightbot–like programming interface	Virtual	6–7	40
Portelance et al. (2016)	Constructing the ScratchJr programming language in the early childhood classroom	ScratchJr	Virtual	5–7	62
Papadakis et al. (2016)	Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study	ScratchJr	Virtual	4–6	43
Hu et al. (2015)	Strawbies: Explorations in tangible programming	Strawbies	Hybrid	4–10	No info
Wang et al. (2014)	A tangible programming tool for children to cultivate computational thinking	T–maze	Hybrid	5–9	20

4.2.2.2. Scratch Jr and Others

Most of the other evaluations involved Scratch Jr. (Papadakis et al., 2016; Portelance et al., 2016; Strawhacker et al., 2018; Pinto and Osório, 2019) and did not evaluate children's use or understanding of control structures, even though the tool enables the use of control structures. The same happened with evaluations of other systems (Wang et al., 2014; Hu et al., 2015; Rose et al., 2017; Jung et al., 2019; Pila et al., 2019; Arfé et al., 2020; Çiftci and Bildiren, 2020). The evaluations that did include reports on the use of control

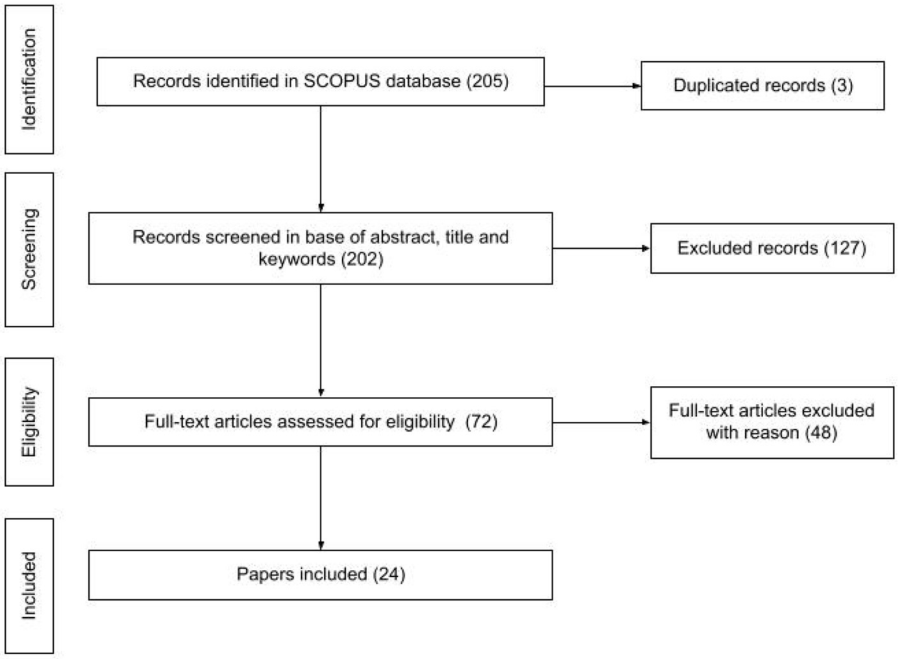


FIGURE 4 | Steps of the selection process of the second SLR. Reported in line with the PRISMA statement (Moher et al., 2009).

structures, without an evaluation, involving Scratch Jr., reported either little use or difficulty with control flow blocks (Pugnali et al., 2017; Strawhacker and Bers, 2019). Another included children in our target age, but also older children without separating their performance (Pugnali et al., 2017). One evaluation of LEGO WeDo found some success with repeat loops, but greater success with CHERP (Strawhacker and Bers, 2015).

4.2.2.3. Bottom Line

Only one study (Sullivan and Bers, 2018) provides evidence of children in Kindergarten mastering conditionals and sensor loops. Multiple studies provide evidence of children in our target age group mastering the use of simple repeat loops (repeat a given # of times) or wait for clap programs. The caveat with all these studies is that they are all from the same research group, use the same system, and the same assessment.

With other tools, except for a study of Lego WeDo which also included CHERP (Strawhacker and Bers, 2015), there are no specific assessments of control flow, other than reports of low use or difficulty with using control flow structures for children in our target age range. In other words, in spite of the great diversity of options for children in our target age range to learn about control flow structures, in our review we found only one technology for which there have been multiple empirical studies to understand whether these children can learn how to use these features.

5. LIMITATIONS

Although we tried to carry out our study in a systematic way, document all the decisions, and report doubtful cases, the current study still has certain limitations. To complement the tools characteristics related to control structures and cost, we had to appeal to online information. We firstly reviewed official websites and online user manuals, but in some cases the information contained in these sources was not sufficient to answer our research questions. In those cases we reviewed unofficial sources such as youtube videos, blogs and private web pages to complete the missing information. We understand that these are not the most convenient information sources, but we used them if there was no available information through official channels. Another limitation related to our online search is that we reported information that we were able to find, which does not ensure that it is the complete existing information. For example, we reported that the application The Foos allows users to build conditionals of "Free condition building" type based on a youtube video that we found, but we cannot ensure that the tool does not allow building other types of conditionals. There is no free online manual that could provide required information, so to confirm that "Free condition building" is the only type that the tool supports it is necessary to pass all the levels that the game offers, and it was impossible for our team to acquire and personally analyze all the relevant tools. Also, our initial list of tools for young children is limited to the tools reported in scientific publications. It is possible that there are valuable tools that were not mentioned in reviews that we analyzed. We tried to address this issue by adding

3 publications that were not found by SLR and by adding four tools that we found in external sources.

6. DISCUSSION

The present study reviewed the state of the art in the teaching of control structures to young children, specifically preliterate children 3 to 6 years of age. While many of the definitions of CT for young children which gather large amounts of consensus amongst academics describe control structures such as conditionals and loops amongst central aspects of CT (Brennan and Resnick, 2012; Grover and Pea, 2013), how this aspect of CT should be developmentally adapted for young children remains unclear. Our findings suggest there is still a large knowledge gap regarding how children acquire early notions about control structures and what the best tools are to introduce children to these concepts. Despite this, these concepts are often included in the interventions targeted at young children and assessed through specific items in the validated CT tests available for young children (Relkin et al., 2020; Zapata-Cáceres et al., 2020).

Our findings demonstrate that there is a wide variety of technological tools which include robots, virtual applications and hybrids, which aim to teach control structures and are targeted to children of these ages. Thus, we infer it is considered relevant that children acquire these concepts early on. Despite this, our findings regarding the reported classroom based research shows that the specifics of how children learn these concepts through the available tools remains unexplored. None of the systematic review articles we identified presented results that were specific to control structures, instead focusing on broader concepts such as CT (Sullivan et al., 2017), programming literacy (Bers, 2019), or engagement (Pinto and Osório, 2019). Given that CT is an umbrella term which encompasses a wide variety of components such as sequencing, using control structures, abstraction, debugging, amongst others (Shute et al., 2017) we must focus on the specifics of each of them in order to have a better sense of the concept as a whole. This is especially relevant for younger children, as the learning curves for each specific skill might differ with age. So far, we found most of the studies focus on several concepts at once but do not further explore learning outcomes for each activity. Thus, the assessments used were more holistic and successful in detecting general learning and engagement outcomes but lacked information on each of the specific tasks and concepts encompassed. An exception to this general approach was the study reported by Kazakoff and Bers (2014) where they focused specifically on sequencing skills, however we did not find any similar study for the learning of control structures, even though our search targeted this term specifically.

Exploring these aspects is also necessary to determine which approaches provide the adequate affordances to enhance learning of each aspect of CT. For example, in our technology overview we observed several approaches to including the use of control structures in tools, such as interrupting events, active wait, or procedural conditions, however there are currently no studies contrasting the strengths and weaknesses of each of these approaches and whether they produce different results in children's understanding of the concepts. As a result, there is only evidence of one tool successfully enabling children to learn some aspects of control structures, mainly due to a lack of studies on the use of other tools by young children that include an assessment of control structure use or understanding.

Moreover, future studies on specific tools should focus on the feasibility of their inclusion in the classrooms in a scalable way. Specifically, our findings regarding the cost of several robots suggest some of them are simply too expensive to be available to all children in a given school or classroom. In addition, some of these tools are more adequately design for individual at-home use, which hinders group based-activities thus elevates the cost of its use even more. Thus, so far the use of robots in education at a large-scale would a entail substantial investment for administrators and policy makers, a problem which could be partially subsided through the design of tools with a group-based focus.

The results of our systematic reviews therefore are encouraging in terms of the wide range of approaches designed for young children to learn about control structures, but also identify a large gap in that we know very little about which of these approaches may work better, or how to structure their use. There is therefore a need for future research to further explore the strengths and weaknesses of the available approaches and understand the feasibility of their use in a variety of contexts (e.g., individual vs. shared, home vs. school).

7. CONCLUSION

The present work demonstrates that there are many diverse tools to support the development of CT in young children. It seems that both academia and industry have interest in designing approaches to enable young children to develop this so-called twenty-first century skill, as we found through our systematic reviews. Although many existing tools allow children to approach advanced programming concepts such as control structures, it is not clear which tools and activities are the most appropriate for teaching them to the youngest programmers. In order to lay the basis for the future research that targets this gap, we provide a systematic overview of existing tools for preliterate children between the ages of 3 and 6. We developed categories that classify the type and complexity of conditionals and iteration structures and used them to categorize each tool. We also provided graphical examples of conditionals that the tools provide.

The analysis of empirical evidence showed that KIBO/CHERP/KIWI is the only tool that consistently demonstrates positive results in teaching control structures to young children. Other tools in our review have not gone through similar evaluations, making it difficult to reach conclusions about their appropriateness for introducing these concepts. The contrast between the diversity of approaches available and the scarcity of evaluations focused on control structures calls for more research, ideally by groups independent of the tools being evaluated, to compare and contrast these approaches in a variety of contexts (e.g., home, preschool).

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

EB, AG, JH, and GTe contributed to conception and design of the study. EB organized the first draft of the manuscript and extracted the tools. EB, AG, and JH wrote sections of the manuscript. EB and AG analyzed the articles involved in the first SLR and reviewed tools characteristics. GTe and JH supervised the revision process. GTr and KP analyzed the articles involved in the second SLR. KP and JH extracted the data and conducted the thematic analysis. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This review is a part of a Ph.D. project of the EB that was supported by Comisión Académica de Posgrado (Uruguay).

REFERENCES

Arfé, B., Vardanega, T., and Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Comput. Educ.* 148, 103807. doi: 10.1016/j.compedu.2020.103807

Bakala, E., Gerosa, A., Hourcade, J. P., and Tejera, G. (2021). Preschool children, robots, and computational thinking: a systematic review. *Int. J. Child Comput. Interact.* 29, 100337. doi: 10.1016/j.ijcci.2021.100337

Bers, M. U. (2019). Coding as another language: a pedagogical approach for teaching computer science in early childhood. *J. Comput. Educ.* 6, 499–528. doi: 10.1007/s40692-019-00147-3

Bers, M. U. (2020). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. New York, NY: Routledge.

Bers, M. U., González-González, C., and Armas-Torres, M. B. (2019). Coding as a playground: promoting positive learning experiences in childhood classrooms. *Comput. Educ.* 138, 130–145. doi: 10.1016/j.compedu.2019.04.013

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., et al. (2016). *Developing computational thinking in compulsory education-implications for policy and practice*. Technical report, Joint Research Centre (Seville site).

Brackmann, C., Barone, D., Casali, A., Boucinha, R., and Muñoz-Hernandez, S. (2016). "Computational thinking: panorama of the americas," in *2016 International Symposium on Computers in Education (SIIE)* (Salamanca: IEEE), 1–6.

Brennan, K., and Resnick, M. (2012). "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (AERA 2012)* (Vancouver, BC), 1–25.

Ching, Y. H., Hsu, Y. C., and Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends* 62, 563–573. doi: 10.1007/s11528-018-0292-7

Çiftci, S., and Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Comput. Sci. Educ.* 30, 3–21. doi: 10.1080/08993408.2019.1696169

CodeSpark Academy Youtube Channel (2022). *codeSpark Youtube Channel*. Available online at: https://www.youtube.com/watch?v=mSv9tsF3G2M (accessed September 6, 2022).

Elkin, M., Sullivan, A., and Bers, M. U. (2016). Programming with the kibo robotics kit in preschool classrooms. *Comput. Sch.* 33, 169–186. doi: 10.1080/07380569.2016.1216251

Elsevier Scopus (2022). *Elsevier Scopus: The Largest Database of Peer-Reviewed Literature*. Available online at: https://www.elsevier.com/en-in/solutions/scopus (accessed March 20, 2022).

Fagerlund, J., Häkkinen, P., Vesisenaho, M., and Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: a systematic review. *Comput. Appl. Eng. Educ.* 29, 12–28. doi: 10.1002/cae.22255

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., and Carboni, A. (2021). Cognitive abilities and computational thinking at age 5: evidence for associations to sequencing and symbolic number comparison. *Comput. Educ. Open* 2, 100043. doi: 10.1016/j.caeo.2021.100043

Getting Started with Osmo Coding Awbie Manual (2022). *Getting Started with Osmo Coding Awbie*. Available online at: https://assets.playosmo.com/static/downloads/GettingStartedWithOsmoCodingAwbie.pdf (accessed March 20, 2022).

Grover, S. (2018). *A Tale of Two CTS (and a Revised Timeline for Computational Thinking)*. Blog@ CACM.

Grover, S., and Pea, R. (2013). Computational thinking in k-12: a review of the state of the field. *Educ. Res.* 42, 38–43. doi: 10.3102/0013189X12463051

Hickmott, D., Prieto-Rodriguez, E., and Holmes, K. (2018). A scoping review of studies on computational thinking in k-12 mathematics classrooms. *Digit. Exp. Math. Educ.* 4, 48–69. doi: 10.1007/s40751-017-0038-8

Hsu, T. C., Chang, S. C., and Hung, Y. T. (2018). How to learn and how to teach computational thinking: suggestions based on a review of the literature. *Comput. Educ.* 126, 296–310. doi: 10.1016/j.compedu.2018.07.004

Hu, F., Zekelman, A., Horn, M., and Judd, F. (2015). "Strawbies: explorations in tangible programming," in *Proceedings of the 14th International Conference on Interaction Design and Children* (New York, NY), 410–413.

Ioannou, A., and Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: a summary of current research and practical proposal for future work. *Educ. Inf. Technol.* 23, 2531–2544. doi: 10.1007/s10639-018-9729-z

Isnaini, R., Budiyanto, C., and Widiastuti, I. (2019). "Robotics-based learning to support computational thinking skills in early childhood," in *AIP Conference Proceedings* (Surakarta).

Jung, H., Kim, H. J., So, S., Kim, J., and Oh, C. (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* (Glasgow), 1–6.

Jurado, E., Fonseca, D., Coderch, J., and Canaleta, X. (2020). Social steam learning at an early age with robotic platforms: a case study in four schools in spain. *Sensors* 20, 3698. doi: 10.3390/s20133698

Kakavas, P., and Ugolini, F. C. (2019). Computational thinking in primary education: a systematic literature review. *Res. Educ. Media* 11, 2019. doi: 10.2478/rem-2019-0023

Kazakoff, E. R., and Bers, M. U. (2014). Put your robot in, put your robot out: sequencing through programming robots in early childhood. *J. Educ. Comput. Res.* 50, 553–573. doi: 10.2190/EC.50.4.f

Khoo, K. Y. (2020). A case study on how children develop computational thinking collaboratively with robotics toys. *Int. J. Educ. Technol. Learn.* 9, 39–51. doi: 10.20448/2003.91.39.51

Kickstarter (2022). *Plobot | Physical Coding for Ages 4 and up*. Available online at: https://www.kickstarter.com/projects/1365116576/plobot-a-robot-coding-companion (accessed March 20, 2022).

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews, volume 4*. Boca Raton, FL: CRC Press.

Kodable (2022). *Kodable.* Available online at: https://www.kodable.com/ (accessed September 6, 2022).

Lyon, J. A., and Magana, A. J. (2020). Computational thinking in higher education: a review of the literature. *Comput. Appl. Eng. Educ.* 28, 1174–1189. doi: 10.1002/cae.22295

Manches, A., and Plowman, L. (2017). Computing education in children's early years: a call for debate. *Br. J. Educ. Technol.* 48, 191–201. doi: 10.1111/bjet.12355

Martin, C. K., Pinkard, N., Erete, S., and Sandherr, J. (2017). "Connections at the family level: supporting parents and caring adults to engage youth in learning about computers and technology," in *Moving Students of Color From Consumers to Producers of Technology* (IGI Global), 220–244.

Matatalab Sensor Add-on (2022). *Matatalab*. Available online at: https://matatalab.com/en/sensor-add-on (accessed September 6, 2022).

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., and the, P. R. I. S. M. A., Group (2009). Reprint–preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Phys. Ther.* 89, 873–880. doi: 10.1093/ptj/89.9.873

OECD (2019). *Oecd Education and Skills Today*. Available online at: https://oecdedutoday.com/computer-science-and-pisa-2021/ (accessed March 20, 2022).

Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., and Wilensky, U. (2016). *Bringing Computational Thinking Into High School Mathematics and Science Classrooms*. Singapore: International Society of the Learning Sciences.

Papadakis, S. (2020). Robots and robotics kits for early childhood and first school age. *Int. J. Interact. Mobile Technol.* 14, 34–56. doi: 10.3991/ijim.v14i18.16631

Papadakis, S. (2021). The impact of coding apps to support young children in computational thinking and computational fluency. A literature review. *Front. Educ.* 6, 657895. doi: 10.3389/feduc.2021.657895

Papadakis, S., Kalogiannakis, M., and Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with scratchjr in preschool education: a case study. *Int. J. Mobile Learn. Organ.* 10, 187–202. doi: 10.1504/IJMLO.2016.077867

Pedersen, B. K. M. K., Larsen, J. C., and Nielsen, J. (2020). The effect of commercially available educational robotics: a systematic review. *Adv. Intell. Syst. Comput.* 1023, 14–27. doi: 10.1007/978-3-030-26945-6_2

Perkovic, L. (2015). *Introduction to Computing Using Python: An Application Development Focus*. John Wiley & Sons.

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., and Wartella, E. A. (2019). Learning to code via tablet applications: an evaluation of daisy the dinosaur and kodable as learning tools for young children. *Comput. Educ.* 128, 52–62. doi: 10.1016/j.compedu.2018.09.006

Pinto, M. S. M., and Osorio, A. (2019). Aprender a programar en Educacion Infantil: analisis con la escala de participacion. *Editorial Universidad de Sevilla* 55, 133–156. doi: 10.12795/pixelbit.2019.i55.08

Portelance, D. J., Strawhacker, A. L., and Bers, M. U. (2016). Constructing the scratchjr programming language in the early childhood classroom. *Int. J. Technol. Design Educ.* 26, 489–504. doi: 10.1007/s10798-015-9325-0

Pugnali, A., Sullivan, A., and Bers, M. U. (2017). The impact of user interface on young children's computational thinking. *J. Inf. Technol. Educ.* 16, 171–193. doi: 10.28945/3768

Relkin, E., de Ruiter, L., and Bers, M. U. (2020). TechCheck: development and validation of an unplugged assessment of computational thinking in early childhood education. *J. Sci. Educ. Technol.* 29, 482–498. doi: 10.1007/s10956-020-09831-x

Robertson, J., Gray, S., Martin, T., and Booth, J. (2020). The relationship between executive functions and computational thinking. *Int. J. Comput. Sci. Educ. Sch.* 3, 35–49. doi: 10.21585/ijcses.v3i4.76

RoboCupJunior (2022). *Robocupjunior About*. Available online at: https://junior.robocup.org/about/ (accessed March 20, 2022).

Román-González, M. (2015). "Computational thinking test: design guidelines and content validation," in *Proceedings of EDULEARN15 Conference* (Barcelona), 2436–2444.

Román-González, M., Pérez-González, J. -C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Comput. Hum. Behav.* 72, 678–691. doi: 10.1016/j.chb.2016.08.047

Rose, S., Habgood, M. J., and Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electron. J. e-Learn.* 15, 297–309. doi: 10.34190/ejel.15.4.2368

Sanford, J. F., and Naidu, J. T. (2016). Computational thinking concepts for grade school. *Contemporary Issues Educ. Res.* 9, 23–32. doi: 10.19030/cier.v9i1.9547

Saxena, A., Lo, C. K., Hew, K. F., and Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: an exploratory study in early childhood education. *Asia-Pacific Educ. Res.* 29, 55–66. doi: 10.1007/s40299-019-00478-w

Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review* 22, 142–158. doi: 10.1016/j.edurev.2017.09.003

Silva, E. F., Dembogurski, B. J., and Semaan, G. S. (2021). A systematic review of computational thinking in early ages. *ArXiv, abs/2106.10275*. doi: 10.48550/arXiv.2106.10275

Sphero (2022). *Sphero brandfolder Indi*. Available online at: https://brandfolder.com/indi (accessed September 6, 2022).

Stamatios, P. (2022). Can preschoolers learn computational thinking and coding skills with scratchjr? A systematic literature review. *Int. J. Educ. Reform.* doi: 10.1177/10567879221076077

Strawhacker, A., and Bers, M. U. (2015). "I want my robot to look for food": comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *Int. J. Technol. Design Educ.* 25, 293–319. doi: 10.1007/s10798-014-9287-7

Strawhacker, A., and Bers, M. U. (2019). What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children. *Educ. Technol. Res. Dev.* 67, 541–575. doi: 10.1007/s11423-018-9622-x

Strawhacker, A., Lee, M., and Bers, M. U. (2018). Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in scratchjr. *Int. J. Technol. Design Educ.* 28, 347–376. doi: 10.1007/s10798-017-9400-9

Sullivan, A., and Bers, M. U. (2016a). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *J. Inf. Technol. Educ.* 15, 145–165. doi: 10.28945/3547

Sullivan, A., and Bers, M. U. (2016b). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *Int. J. Technol. Design Educ.* 26, 3–20. doi: 10.1007/s10798-015-9304-5

Sullivan, A., and Bers, M. U. (2018). Dancing robots: integrating art, music, and robotics in singapore's early childhood centers. *Int. J. Technol. Design Educ.* 28, 325–346. doi: 10.1007/s10798-017-9397-0

Sullivan, A., and Bers, M. U. (2019). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. *Int. J. Technol. Design Educ.* 29, 1033–1051. doi: 10.1007/s10798-018-9483-y

Sullivan, A. A., Bers, M. U., and Mihm, C. (2017). *Imagining, Playing, and Coding With Kibo: Using Robotics to Foster Computational Thinking in Young Children*. Hong Kong: Siu-cheung KONG The Education University of Hong Kong.

Tang, X., Yin, Y., Lin, Q., Hadad, R., and Zhai, X. (2020). Assessing computational thinking: a systematic review of empirical studies. *Comput. Educ.* 148, 103798. doi: 10.1016/j.compedu.2019.103798

Taslibeyaz, E., Kursun, E., and Karaman, S. (2020). How to develop computational thinking: a systematic review of empirical studies. *Inform. Educ.* 19, 701–719. doi: 10.15388/infedu.2020.30

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M., and Yeo, S. H. (2016). A review on the use of robots in education and young children. *Educ. Technol. Soc.* 19, 148–163.

Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., et al. (2022). A cognitive definition of computational thinking in primary education. *Comput. Educ.* 179, 104425. doi: 10.1016/j.compedu.2021.104425

Tufts University. (2022). *Kibo Robot*. Available online at: https://sites.tufts.edu/devtech/research/kibo-robot/ (accessed March 20, 2022).

Umam, M. U. K., Budiyanto, C., and Rahmawati, A. (2019). "Literature review of robotics learning devices to facilitate the development of computational thinking in early childhood," in *AIP Conference Proceedings*.

Wang, D., Wang, T., and Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking. *Scientific World J*. 2014, 428080. doi: 10.1155/2014/428080

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *J. Sci. Educ. Technol*. 25, 127–147. doi: 10.1007/s10956-015-9581-5

Williamson, B. (2016). Political computational thinking: policy networks, digital governance and 'learning to code'. *Crit. Policy Stud*. 10, 39–58. doi: 10.1080/19460171.2015.1052003

Wing, J. (2011). Research notebook: computational thinking–what and why. *Link Mag*. 6, 20–23.

Wing, J. M. (2006). Computational thinking. *Commun. ACM*. 24, 33. doi: 10.1145/1118178.1118215

Yadav, A., Hong, H., and Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *TechTrends* 60, 565–568. doi: 10.1007/s11528-016-0087-7

Yadav, A., Sands, P., Good, J., and Lishinki, A. (2018). "Computer science and computational thinking in the curriculum: research and practice," in *Handbook of Information Technology in Primary and Secondary Education*, 1–18.

Yang, K., Liu, X., and Chen, G. (2020). The influence of robots on students' computational thinking: a literature review. *Int. J. Inf. Educ. Technol*. 10, 627–631. doi: 10.18178/ijiet.2020.10.8.1435

Yu, J., and Roque, R. (2019). A review of computational toys and kits for young children. *Int. J. Child Comput. Interact*. 21, 17–36. doi: 10.1016/j.ijcci.2019.04.001

Zapata-Cáceres, M., Martín-Barroso, E., and Román-González, M. (2020). "Computational thinking test for beginners: design and content validation," in *2020 IEEE Global Engineering Education Conference (EDUCON)* (Porto: IEEE), 1905–1914.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX

TABLE A1 | Search term used with each tool to search in SCOPUS.

Tool name	Search term	Search results	Relevant results
Bee Bot	TITLE-ABS-KEY (((Bee bot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	23	
Blue Bot	TITLE-ABS-KEY (((Blue bot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	3	
Botley	TITLE-ABS-KEY (((botley AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	
Code-a-Pillar	TITLE-ABS-KEY ((({Code-a-Pillar} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Cubetto	TITLE-ABS-KEY ((({Cubetto} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	7	
KIBO	TITLE-ABS-KEY ((({KIBO} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	20	8
KIWI	TITLE-ABS-KEY (((kiwi AND robot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	2	2
KUBO robot	TITLE-ABS-KEY (((kubo AND robot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Matatalab Coding Set	TITLE-ABS-KEY (((Matatalab AND robot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
mTiny	TITLE-ABS-KEY (((mtiny AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Ozobot Evo	TITLE-ABS-KEY (((ozobot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	6	
Ozobot Bit	considered above		
Plobot	TITLE-ABS-KEY (((plobot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Pro-bot	TITLE-ABS-KEY (((pro-bot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Qobo	TITLE-ABS-KEY (((qobo AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Robot Mouse	TITLE-ABS-KEY ((({Robot Mouse} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Robotito	TITLE-ABS-KEY (((robotito AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	2	
Sphero indi	TITLE-ABS-KEY ((({sphero} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	15	
TurtleBot	TITLE-ABS-KEY (((Turtlebot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	4	

(Continued)

TABLE A1 | Continued

Tool name	Search term	Search results	Relevant results
VEX 123	TITLE-ABS-KEY ((({vex 123} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
BOTS	TITLE-ABS-KEY ((({BOTS} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8)))) AND (EXCLUDE (SUBJAREA , "MEDI"))	39	1
Codeable Crafts	TITLE-ABS-KEY ((({Codeable Crafts} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Code.org	TITLE-ABS-KEY ((({code.org} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	19	2
Kodable	TITLE-ABS-KEY ((({Kodable} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	3	1
LightBotJr	TITLE-ABS-KEY ((({LightBotJr} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Move the turtle	TITLE-ABS-KEY ((({move the turtle} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	1
RoboZZle	TITLE-ABS-KEY ((({RoboZZle} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Run Marco!	TITLE-ABS-KEY ((({Run Marco} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	2	
ScratchJr	TITLE-ABS-KEY ((({ScratchJr} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	28	7
The Foos	TITLE-ABS-KEY ((({The Foos} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Tynker: Coding for Kids	TITLE-ABS-KEY ((({Tynker} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	
Blue Bot	Repeated tool		
CHERP	TITLE-ABS-KEY (((cherp AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	4	2
COJI	TITLE-ABS-KEY ((({coji} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Dash and/or Dot	TITLE-ABS-KEY ((({Dash} AND robot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	3	
Finch	TITLE-ABS-KEY ((({finch} AND robot AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	
Sphero indi	Repeated tool		
Thymio	TITLE-ABS-KEY ((({Thymio} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	13	
Tinkerbots	TITLE-ABS-KEY (((tinkerbots AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	

(Continued)

TABLE A1 | Continued

Tool name	Search term	Search results	Relevant results
Makeblock Neuron	TITLE-ABS-KEY (((makeblock AND neuron AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Puzzlets Starter Pack	TITLE-ABS-KEY ((({Puzzlets} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Roberto	TITLE-ABS-KEY ((({Roberto} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8)))) AND (EXCLUDE (SUBJAREA,"MEDI") OR EXCLUDE (SUBJAREA,"NURS") OR EXCLUDE (SUBJAREA,"NEUR") OR EXCLUDE (SUBJAREA,"PHAR") OR EXCLUDE (SUBJAREA,"IMMU") OR EXCLUDE (SUBJAREA,"BIOC")) AND (EXCLUDE (SUBJAREA,"ARTS") OR EXCLUDE (SUBJAREA,"SOCI"))	5	
Scottie Go	TITLE-ABS-KEY ((({Scottie Go} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	0	
Coding Awbie	TITLE-ABS-KEY ((({strawbies} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	1
Tabletop puzzle block system	TITLE-ABS-KEY ((({Tabletop puzzle} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8))))	1	
T-Maze	TITLE-ABS-KEY ((({t-maze} AND (preschool OR child OR {early age} OR kindergarten OR {lower education} OR {early years} OR {elementary education} OR {young learner} OR {primary school} OR {primary education} OR k-6 OR k-8)))) AND (EXCLUDE (SUBJAREA , "BIOC") OR EXCLUDE (SUBJAREA , "MEDI") OR EXCLUDE (SUBJAREA , "PHAR")) AND (EXCLUDE (SUBJAREA , "NEUR"))	1	1

Appendix 5

“It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers

Ewelina Bakala et al. «“It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers». In: *Proceedings of the 2023 ACM Conference on Information Technology for Social Good.* 2023, pp. 477–486

Author’s contribution The author conceptualized the research idea, designed the methodology, and developed the data collection strategies. She participated conducted the focus groups and, along with another researcher, performed the thematic analysis. She also led and actively participated in the writing process.

“It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers

Ewelina Bakala
ebakala@fing.edu.uy
Instituto de Computación, Facultad de Ingeniería,
Universidad de la República
Uruguay

Ana Crisitna Pires
ana.pires@iti.larsys.pt
Interactive Technologies Institute, Universidade de Lisboa
Portugal

Gonzalo Tejera
gtejera@fing.edu.uy
Instituto de Computación, Facultad de Ingeniería,
Universidad de la República
Uruguay

Juan Palo Hourcade
Department of Computer Science, University of Iowa
USA
juanpablo-hourcade@uiowa.edu

ABSTRACT

This paper presents a study with kindergarten teachers to assess the advantages, challenges and opportunities of commercial robots to teach computational thinking to young children. Recent studies have highlighted the potential benefits of introducing CT concepts at an early stage. Robots are an engaging and effective educational tool for teaching CT to young children, providing hands-on and interactive learning experiences. Entirely tangible robotic environments have successfully connected the abstract world of CT with the concrete world of preschoolers. Children can program robots by pressing buttons, drawing the path or using code cards. However, there is limited research on the use of commercial robots in preschool classrooms. This research aims to address this gap by investigating preschool teachers’ perspectives on the advantages, challenges, and opportunities associated with using commercial robots in the context of kindergarten classrooms. We contribute with a list of practical, pedagogical and motivational aspects that should be taken into account while evaluating robots and design considerations to build robotic environments for kindergarten classrooms.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**; • **Social and professional topics** → *Children.*

KEYWORDS

Educational robotics, Preschoolers, Teachers’ perspective

ACM Reference Format:
Ewelina Bakala, Ana Crisitna Pires, Gonzalo Tejera, and Juan Palo Hourcade. 2023. “It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers. In *ACM International Conference on Information Technology for Social Good (GoodIT ’23), September 6–8, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3582515.3609570

GoodIT ’23, September 6–8, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0116-0/23/09…$15.00
https://doi.org/10.1145/3582515.3609570

1 INTRODUCTION

In today’s rapidly advancing technological landscape, computational thinking (CT) has emerged as a crucial skill for individuals of all ages. Defined as an approach that uses computer science concepts to solve problems [28, 29], CT plays a pivotal role in using computers as a creative tool and supporting problem-solving in the digital age. By integrating CT into school curricula, educators can foster critical cognitive abilities, including abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization [7]. While traditionally perceived as a domain for older students, recent research has highlighted the potential benefits of introducing CT at an early age, particularly during the preschool years [21, 23, 26, 27].

Among the various educational tools available, robots have garnered attention as effective vehicles for teaching CT to young children. They offer a tangible and interactive learning experience that captivates preschoolers’ imagination and engages them in the learning process. The physical presence of robots provides a unique advantage over traditional educational approaches by enabling hands-on exploration and experiential learning.

Tangible user interfaces (TUIs) for programming, in particular, have resulted as an appropriate method for introducing CT concepts to young children. These interfaces utilize physical objects such as tiles, blocks, or cards that children can manipulate and arrange to create simple programs. By associating physical actions with coding concepts, TUIs bridge the gap between the abstract world of CT and the concrete world of preschoolers, making the learning process more accessible and enjoyable. Empirical studies have provided evidence that interventions with robots programmed using TUIs were associated with preschoolers being able to improve skills such as sequencing [1, 6, 10, 11, 16, 17, 22, 25, 26], problem solving [17, 25], debugging [1, 6, 12, 16, 22] and even effectively employing control structures (conditionals and loops) [6, 22] that are essential for the construction of advanced algorithms [15].

Despite the proliferation of commercial robots on the market, most of these devices are designed for individual use, focusing primarily on entertainment. Consequently, there is a lack of research and development surrounding the use of robots in a preschool classroom setting, considering use by groups of children. As we mentioned before, many articles report the use of commercial robots for preschoolers in empirical studies. However, evaluating the robot's appropriateness for classroom activities is almost never the focus of those scientific communications. In a few cases, there are specific comments on child-robot interaction; for example, "authors hypothesized that the use of an external memory system for keeping a visual record of the commands used to program the Bee-Bot would be necessary for effectively scaffolding children's learning" in [1] or "even though we had gone through this set of activities with our K1 students (aged 3 to 4), they did not fully comprehend those vocabularies/instructions (e.g., turn left/right) used in the Bee-Bot activity" in [26]. The studies generally focus on one particular robot and evaluate its effectiveness in CT development. If present, the observations related to the interaction are made by the researchers conducting the activities. In our previous work [3], we evaluated multiple robots in a classroom setting, but the observations were made by the authors and focused on group interaction. To our knowledge, none of the previous studies sought the views of teachers on multiple robots for preschoolers.

Recognizing this gap, we evaluated four commercial robots with two preschool teachers to gather their perspectives on the use of robots in their classrooms and their potential for effectively teaching programming concepts such as sequences and control structures. The following research question guided this work: What are the perceived advantages, challenges, and opportunities of commercial robots to be implemented in a preschool classroom, according to teachers?

2 METHODOLOGY

Between March and May 2023, we conducted five focus groups with two preschool computing teachers from a private educational center in Montevideo. Both teachers have more than ten years of teaching experience in public and private institutions and work at the preschool and primary school level. The study protocol was approved by the Ethics Board of the lead institution and all methods were in accordance with the Declaration of Helsinki [2].

2.1 Robots

We evaluated four tangible off-the-shelf commercial robots: Qobo [24], Ozobot [19], KIBO [13] and Botley [14]. We selected them as they can be programmed using tangible interfaces and offer the possibility to program with control structures.

Qobo is a snail-like robot with two acting modes: game mode and free mode. In the game mode, children connect tangible puzzle-shaped cards to guide the robot from the start position (game mode card) to the destination card (gem card). The robot senses the cards and moves according to the instruction associated with each card. Free mode enables users to scan the cards and execute the stored program without coding cards below the robot. The conditional card (banana left, apple right) allows directing the robot left or right according to the previous input (banana or apple card).

Ozobot follows a black line and responds to color codes composed of three colors [18] with changes in its behavior. The color codes can change the robot's speed, start a special movement (like zig-zag or spin) or define the robot's direction in the next bifurcation (left, straight, or right).

KIBO can be programmed by scanning barcodes printed on wooden blocks used to build the sequence of orders. Depending on the kit, it can include different sensors and actuators. It counts with an if-block that can be combined with "near," "far," "light," and "dark" conditions.

Botley comes with two modes: "line" and "code." In the "line" mode, it follows black lines; in "code" mode, it can be programmed using a remote control. The child can press buttons on the remote control that define the robot's sequence of actions (main program) and send the program to the robot with the green "transmit" button. Conditionals can be implemented by defining actions the robot will execute if an obstacle is detected (conditional program).

2.2 Procedure

We evaluated one robot per session with two teachers (sessions 1-4). Sessions started with a brief introduction of the robot, its functions, and how to program it. Then, teachers explored it independently, prepared simple programs, and commented on their impressions about the robot. To further fuel the discussion, we asked teachers to point out the advantages, challenges, and opportunities each robot has in their opinion and how they envision using them in their kindergarten classrooms.

The sessions were video recorded, and two researchers performed a reflexive thematic analysis [8] of the videos from the focus group with teachers. We followed a mixed coding approach, where we designed the first codebook and inductively extended it with observed codes. We went back to teachers to triangulate our results and enrich our analyses. We presented the analyzed themes to confirm our findings' correctness, to consult items we had doubts about, and to gather new feedback after classroom activities that the teachers implemented with Ozobot, Qobo, and KIBO (session 5). Results from session 5 allowed us to have a more profound understanding of their opinion and identify new relevant aspects that emerged during classroom activities.

3 RESULTS

During the thematic analysis, we identified three relevant themes: practical (e.g., size, battery duration, fragility), pedagogical (e.g., concepts that can be explored, appropriateness of the programming interface), and motivational (e.g., attractive design or children's interests) aspects. We used these themes to classify teachers' comments. Here, we present the results of the evaluation of each robot (see Table 1), a summary of relevant items that can be evaluated in robots in general (see Appendix A), and considerations related to the use of the robots in a classroom context.

3.1 Robots' Evaluation

Each robot was analyzed considering the comments in all five focus groups.

3.1.1 Qobo. Many practical aspects were mentioned as Qobo's strengths. Both teachers, T1 and T2, considered that its size and shape were appropriate for young users because they could lift it with just one hand and grab it easily due to its form. Both observed that the robot was very precise in its movements while executing

		Strenghts	Weaknesses	Opportunities
Qobo	practical	size form battery charging battery duration movements precision	step size expensive to fix and mantain inexpected bahaviours	
	pedagocical	no previous work required program-robot distance errors detection multimodal output guidance during the activity interaction with the child	interaction with sensors no loop in play mode limited loop in free mode conditional too rigid amount of conditionals confusing mat	mat for loops accessibility of coding cards connection with teaching curriculum
	motivational	interaction with the child	little innovative	
Ozobot	practical	size	movements precision color codes detection battery duration form fragility expensive to fix and mantain color codes drawing	simplify color codes drawing change color codes to icons stickers with color codes
	pedagocical		color codes complexity target age collaboration	codes with inverse reading previous work with color codes connection with teaching curriculum
	motivational	innovative fun codes design		
KIBO	practical	coding blocks size coding blocks form	size fragility color of the light blocks no color relation between blocks and sensors/actuators program uploading program decomposition while uploading amount of programming blocks precision of light/dark concept expensive to fix and mantain	program uploading
	pedagocical	advanced programming synatx (begin-end) interaction with sensors control over diverse actuators unique evaluation of if-statement	program-robot distance	suitable for older children connection with teaching curriculum
Botley	practical	size roboust auxilary cards - colors program uploading extension of the uploaded program	size fragility auxilary cards - time auxilary cards - easy to disorder step size velocity expensive to fix and mantain	
	pedagocical	auxilary cards - visibility of the program	program-robot distance no program visibility auxiliary cards placement loops - sintaxis conditionals - sintaxis conditionals - prediction of the secuence conditionals - cognitive demand auditive feedback	connection with teaching curriculum
	motivational	design		

Table 1: Relevant practical, pedagogical and motivational aspects for each robot.

the program. They also considered that charging its battery via microUSB was very practical.

From a pedagogical point of view, T2 was enthusiastic about the vocalization that is used to reinforce the robot's actions (for example, it says "forward!" when it passes over a coding card that makes it move forward) and to guide the programming activity (it provides audio clues indicating where to put the robot and how). She considered it beneficial that "it reinforces the visual output with audio [...] it (the information) enters by two senses" and, by guiding the child, "promotes autonomy". Coding cards in puzzle form were considered a clear and direct programming interface. Both teachers agreed that they helped to visualize the program; T2 mentioned that "the sequence is visually explicit" and that the interface "is super clear when programming an algorithm." Many times during the focus groups, T2 mentioned program-robot distance as an important aspect at the preschool level, in case of Qobo she considered that the distance is very low as the robot moves over the programming cards and the child can easily follow the program execution. She also considered the puzzle shape of the cards helpful while preparing the program. The puzzle form indicates where the following command should be attached, and so prevents programming syntax errors. Cards that require a child's action (jiggles or lifting the robot) were polemic. T1 considered that they "create a motivation," "reinforce what a command is," and "help children to focus on the program execution," as the robot does not execute the following steps if the child does not interact with it. T2 appreciated its motivational aspects but doubted its pedagogical use.

Although the coding puzzle cards were positively evaluated as a programming interface, their size was considered limiting, as they did not allow building long and complex paths with children working at the classroom tables. Also, T2 mentioned that the cards could have tangible clues for children with low vision or blind. The teachers were also not satisfied with the implementation of the loops as the free mode has only a fixed number of repetitions (3 or 6 times), and the game mode did not support loops. T2 proposed a mat with repeated patterns (for example, concatenated L-shaped road units) that could support the teaching of the loop concept. Also, the implementation of conditionals was considered not very challenging and too rigid and the teachers expressed the desire to be able to work with more diverse conditionals.

Although the teachers noticed some unexpected behaviors and considered the robot expensive to fix and maintain (like all robots in Uruguay), they saw many possibilities to connect programming activities with other curricular contents. T1 stated that "it has many possibilities" and "you can integrate it with a lot of things" and named mathematics (counting, sequencing, geometry, probability, magnitudes, sets), spatial positioning, and social bonding.

Qobo was the first robot discussed in a focus group; T2 used it in informatics classes with children of level 4 (3 to 4 years old) after the focus group. T1 also assisted in some of those instances, and both shared their observations in the final focus group. They agreed that Qobo does not require preliminary work with children, contrary to other robots analyzed in the focus groups. They were surprised by the battery duration ("I never charged it!" stated T2) but disappointed with the card materials as some card tips began to peel off. They complained about the mat because the children were too influenced by its design (they tried to follow the painted roads with the path they were building and avoided places where water was drawn, see [24] to consult the mat design). T1 preferred a clean white mat with a grid, "I do not like anything that structures it so much," she said and claimed that too much structure limits the activities and makes it difficult to work with the robot over a sustained period of time.

T1 observed that although the puzzle-shaped form had the potential to prevent errors, some children ignored that the cards should be connected to each other and committed programming errors anyway. They thought it could be beneficial to have both puzzles that define the place to concatenate the following command (current version) and cards with no obvious place to continue the program. They proposed square-shaped cards with no inserts or cards with multiple inserts that allow concatenating commands in incorrect positions. They considered that they could be more challenging for older children and allow them to learn from errors.

The opportunity to see other robots (specially KIBO) made them notice that Qobo's interaction with the environment using sensors is limited and could be extended.

In general, they found Qobo very appropriate for the preschool level. However, they admitted that it is not innovative and has a "low ceiling" and that it would be difficult to use it over a sustained period of time.

3.1.2 Ozobot. Its small size and transparent body which allow children to observe its circuits inside were mentioned as aspects that makes it curious and attractive. Before using Ozobot with children, teachers mentioned that the size could be a practical weakness as it seems too small and fragile "I do not see it as robust, I am afraid that it could fall down and 'puff' [does not work anymore]" - T2. However, after using it, they mentioned that the weakness was not the size but its shape. They suggested that a more secure casing or shape could prevent the robot from falling.

A motivational aspect, much appreciated by the teachers, was the possibility to draw its path: "Hand tracing has a relationship with art that I like! It's free, it's innovative and creative, and it connects with other things about the child, previous experiences, and that makes them more enthusiastic [...] I like it with markers instead of cards. I think this can be more open, and children may be more involved in the design" - T2. Also, some features and codes such as "tornado" (the robot spins) and "turbo" (the robot changes its speed to go super fast) were considered motivational factors that could engage and motivate children to play with the Ozobot. The teachers identified several opportunities for using it in the classroom. The use of codes and observing Ozobot's behavior was seen as a way to develop logical reasoning and work on various skills such as path recognition, serialization, directionality, and mathematical concepts like sequences, perimeter, and amplitude. The teachers believed that Ozobot had the potential to remain relevant and not become deprecated in terms of its didactic aspects.

The teachers evaluated Ozobot's weaknesses and commented on practical and didactic aspects. Regarding the didactic aspects, T2 questioned us about the complexity of the options to program the robot: "Why did they [the developers] choose color codes instead of using icons?" She was negatively surprised about this limitation. She also mentioned that so many color codes would be confusing for children and that limiting the number of color codes to three would be better. Also, T2 expressed concern about the difficulty children might face in accurately painting the color code in the black line that Ozobot follows. To address concerns about using color codes with young children, the teachers made several suggestions, such as using small squares or stencils for children to paint inside, making the process more manageable, stickers with codes or paths, including curves and straight lines, etc. T2 proposed making the colors more similar to icons to help children focus on the symbol's meaning rather than memorizing abstract color associations.

T2 also mentioned the challenge of precise line drawing for the robot "is very difficult for children. It is not the instruction per se but the instruction format.[...] it is about how children draw and the possible challenges for the robot as their lines and drawings are imprecise." In terms of practical aspects, the teachers mentioned issues with the reading sensor and battery autonomy. After using it with children, the teachers observed that it struggled to distinguish between black and blue lines under certain lighting conditions or when the robot had a low battery. This practical weakness was further exemplified when the Ozobot, after performing several spins during the "tornado" function, often ended up off the line and could not continue its intended path. T1 commented: "It was very difficult to draw the circuits - you have to explain a lot of things - the lines could not be wide or thin, the color intensity, when the battery is running out it makes mistakes. [But because of that] we started to talk a lot about the mistakes. They [the children] started to realize it, that the color sensor was failing." Despite these practical challenges, the teachers appreciated how these issues prompted discussions about the robot's limitations and encouraged students to recognize them.

The teachers also stressed the importance of providing prior training to children to understand robot's responses to the codes and the idea of color patterns that codify actions. They suggested creating a path on the floor and make children follow the path and simulate robot’s responses and a pattern recognition activities in which one child uses a secret code to send a message that the other child should try to discover. To encourage collaboration, the teachers suggested that drawing activities could provide children more opportunities to work together than using separate pieces, as seen with Qobo. They recommended using larger sheets to accommodate the robot’s trajectory to allow for collaborative work: “the collaboration comes from making the drawings. It could be more collaboration than using pieces, as with the Qobo. Because the robot is so small, its trajectory could be big. We could use a big sheet so they could collaborate.”- T2. Overall, the teachers recognized the strengths of Ozobot, such as its ability to motivate and engage children, the opportunity to draw paths, and the potential for various learning activities. They also identified areas for improvement, particularly in simplifying the color codes and addressing practical issues with sensor reading, battery autonomy, and fragility. The teachers envisioned strategies for collaboration and provided suggestions for using Ozobot effectively with young children, including prior training and incorporating more accessible elements like icons or stickers.

3.1.3 KIBO. T2 appreciated KIBO’s design, mentioning that it aligned well with Waldorf’s pedagogy [5]. In terms of practical aspects, they mentioned the blocks’ size, the “begin” and “end” commands, and how they fit together. They liked how the blocks fit together easily, making them suitable for young children. T2 said, “I like how they fit together, their size... Even kids three years olds can do it.” She also mentioned that the “begin” and “end” blocks allow children to easily understand where the sequence starts and ends and also help children to get familiarized with advanced programming syntax, similar to actual coding.

In terms of didactic aspects, T1 praised KIBO’s sensors and actuators, considering them comprehensive and exciting: “I am excited; this is so complete. And it has several actuators.” Teachers preferred KIBO’s if-statement evaluation, which occurs only once, compared to Botley’s continuous evaluation, which makes the robot’s behavior difficult to predict.

The greatest KIBO weaknesses and threats detected were related to practical aspects. The most mentioned negative practical aspect was related to uploading the created programs, which was deemed difficult and not child-friendly: “I do not like this part, it is not for children, will not be easy for children”- T2 and T1 added: “to upload the program we should upload the blocks one by one, which is difficult with children [...] otherwise children would upload the nearest blocks, and it would be very confusing.[...] I really like KIBO, but if we cannot upload the program, I cannot use it either!”- T2. When the teachers interacted with KIBO, they struggled to upload the program. They even made an analogy with self-checkout kiosks at supermarkets, which are also difficult sometimes. T2 mentioned, “I wouldn’t mind if the child did the sequence and I loaded the sequence... But if I can’t do it either… I would not use this for preschoolers because the reading instructions would generate a lot of frustration and little self-regulation.” And T1 said: “I also got frustrated [not only the kids].” Another negative aspect not appreciated by the teachers was the design of KIBO. It was considered too big and fragile.

After using it with children, the teacher reinforced some of their previous expectations with KIBO. The teachers did not like KIBO’s design inconsistencies, such as no color match between the coding blocks and sensors and actuators. Also, the blocks related to light control were confusing as the background color has more presence than the color of the icon, which indicates what color the light will be. Also, the light icon was confusing (e.g.: “the light seems like a spider without two legs”- T1).

A negative didactic aspect was related to the program’s location outside the robot and the need to decompose it hindered understanding and execution. Also, there was the need to decompose the program (blocks fitted together) to scan one by one. So, when grabbing each block to put on the front of the scanner, they did not fit it again when returning the block to the table, making it hard to understand the program and the robot’s execution.

The “if” statement was initially thought to be challenging for children but was found to be understandable after using KIBO. However, other negative issues emerged after using it with children, such as the limited quantity of directional blocks. The “if dark” statement was unclear in its operation.

The teachers detected opportunities connected to the curriculum. Some motivational aspects were mentioned, such as that KIBO could be integrated into the curriculum, connecting with information technology concepts and storytelling (a motivational aspect). For instance, if it is dark, the robot could turn the light on (use a light sensor). Suggestions were made for additional features, such as incorporating a pencil: T2 asked, “could we add a pencil? [after KIBO made a path in the form of a square] and the square is visually captured.” When we presented a little sign that can be used at the top of KIBO, T2 said: “it would be great if it could take a message to another child at another table,” enabling message delivery between tables. Finally, teachers considered KIBO suitable for older children, even in third and fourth grade.

3.1.4 Botley. The most discussed aspects of Botley were the auxiliary cards and the implementation of control structures. Auxiliary cards are paper cards with color arrows that indicate Botley’s four movement directions and are used to visualize the program before uploading it to the robot using the remote control. T1 liked the idea of first thinking and preparing the sequence and then uploading the program. She also appreciated that the colors of the arrows on the cards match the colors of the remote control buttons, making the program upload very easy: “I really like the card with the color because it does not give me much chance to commit errors.” Remote control as a programming interface was considered simple and fast, but the teachers admitted that, if using only remote control without auxiliary cards, it was difficult to visualize the program. T2 previously worked with Botley and was not so enthusiastic about the auxiliary card: “In the end, I do not use them because it takes too much time” and that “they are not for classroom use” as it is easy to mess them up accidentally. She also did not like that they increase the distance between the program and the robot - the child has to first prepare the cards, then use the remote control to upload the program, and then the robot executes the corresponding actions: "There are three steps; the child got lost [...] it has to be more instantaneous."

How to order the auxiliary cards on the table while preparing the program was also an issue that underwent heavy discussion and none of the options seemed to satisfy both teachers. For T2 putting one next to the other, from left to right, was a convincing option. "For me, the best thing is to put it like this, as you read a story, a word," but T1 was more keen to put the following arrow command at the end of the current, simulating the robot's movement in space. This spatial placement did not fully represent the corresponding robot movements as the robot rotates in place, and the arrow on the turning card gives the impression that the robot will move to the side.

Also, the cards' order to represent loops was not convincing. As the loop button is used to start the loop and repeat it (there is no numerical parameter to indicate the number of repetitions) and there is no end-loop command, the teachers found it difficult to visually order the cards and explain how the commands will be executed. T2 found that "The way in which it should be entered is not the way in which the child can reason about it and be clear as to what will happen" and was doubting whether it was appropriate for kindergarten "It says it is for children aged 5; I do not think that a 5-year-old child can do it". T1 also noticed that due to the syntax, it is impossible to execute one loop and immediately the second one as Botley interprets it as loop - commands - start loop, instead of loop - loop.

The same problem with the visual representation of the program was present while working with conditionals. The conditionals are associated with object detection, and the commands associated with it can be executed at any moment of the main program and more than once. The robot is constantly sensing the environment, and the conditional commands can be executed at the beginning of the programming step, in the middle, and even after the whole program is executed. The impossibility of predicting when the object will be detected made it difficult to visually represent the sequence of actions the robot will take. T1 considered the program's syntax confusing as it did not reflect the sequence of the robot's actions.

Both teachers complained that they can not prepare a program if they do not know when the obstacle will be detected. T1 said, "Yes, it is difficult to have a conditional and not know what obstacles it detects and when; it is also difficult to see if you have executed the entire program". T2 tried to think of an exercise in which the robot goes from A to B in the grid using conditionals but was not able to combine the main program (moving forward) with object detection as she was not sure if the obstacle in front of the robot will be detected during the first forward step or at the beginning of the second one: "What happens here is that it is not just 'always forward,' you have to put how many times." They were complaining that the robot "does not do the same thing twice" (T1) and "it prevents me from reaching my goal" (T2). T1 commented, "The problem with this is that with all the kids working around it, all the time, it's going to be detecting things in front." Both were not able to come up with a reasonable example of a problem that could be resolved using conditionals. T2 stated, "I do not know how to use it with conditionals." She also considered that it is complex and too demanding for preschoolers to prepare and follow two parallel programs (main and conditional program).

Botley stores the last uploaded program, and pressing directional buttons after uploading adds new commands to the current program. The teachers liked this possibility, although T2 said that when working with preschoolers, she always asks them to start the program with the trash button that removes the previously uploaded program and then upload the program from the beginning.

With respect to its size and fragility, the teachers considered it the correct size but did not like that children need to use two hands to lift it. T1 considered that it "seems quite robust," but T2 saw it as fragile, as she had already discussed a classroom accident in which the robot fell down and its wheel stopped working.

Both agreed that the robot moves too fast, making it difficult for children to follow the uploaded program. They missed audio feedback reinforcing the robot's actions, and its steps were considered too big. T2 complained that if you want to count up to 10 (10 movements in a straight line), it will take too much space.

Like all previously evaluated robots, they considered it expensive to fix and maintain and saw multiple opportunities to connect programming activities with other curricular contents.

3.2 Relevant Aspects

We identified diverse practical, pedagogical, and motivational aspects related to the robots' classroom use (see Table 1). We summarized them to provide future research with a list of items relevant for teachers in the classroom context. We adapted robot-specific items (for example, "number of coding blocks" in the case of KIBO) to more general aspects that can be evaluated in robots in general ("number of coding elements"). Some items (for example, auxiliary cards for Botley) were so robot-specific that they could not be generalized and were left apart. In the Appendix A, we present the items grouped by category and a scale to evaluate them using, for example, questionnaires.

3.2.1 General considerations. Many general aspects mentioned by the teachers are relevant when working with robots and children. Available time was a crucial variable to plan activities and define the size of the group. Both teachers stated that working in very small groups (2 to 3 children) or individually is always better. T2 stated, "With the youngest, the fewer, the better" - T2. But both admitted that they usually work with bigger groups due to time constraints. They considered that having more robots would not help provide a better educational experience as the activities with robots require constant supervision and mentioned the "rotative tables" as a strategy they apply to work with smaller groups with constant supervision. They separate children into groups, and each group works at a different table. Some tables do familiar activities that the children can do independently (draw, play with blocks, or on tablets), and one table works with robots under the teacher's supervision. The children rotate so that all of them pass through all tables. T1 mentioned that sometimes not all the children are able to participate in activities with robots, and some groups work with them in one session and others in the following one, and the children are flexible and have no problems accepting the situation. Regarding the area to work with robots, T2 highlighted that she prefers to work on the floor, while T1 preferred to work at the tables: "It does not work for me on the floor; they go all over the place".

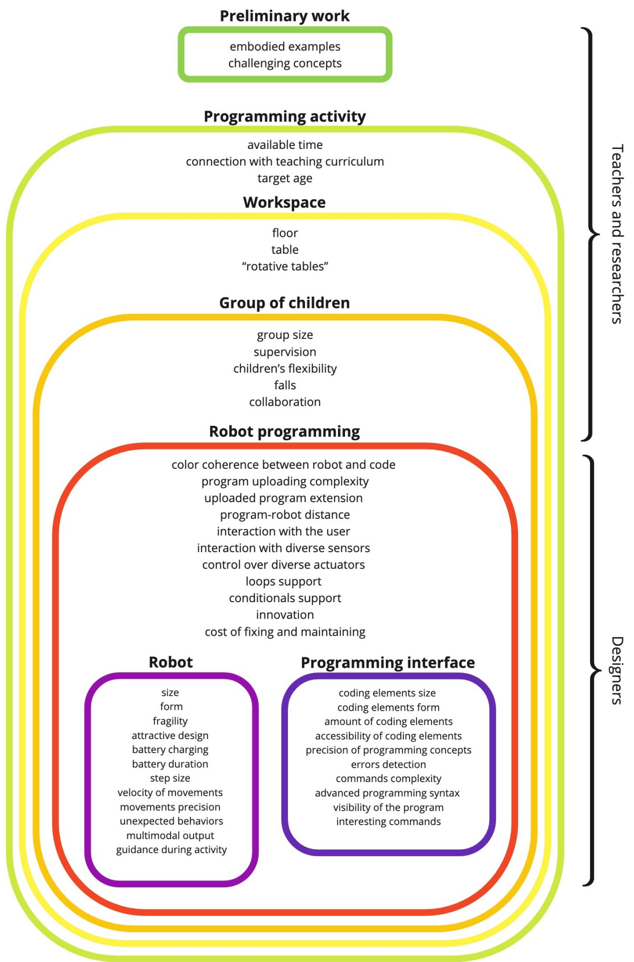


Figure 1: General considerations and relevant aspects grouped by activity level. We divided the identified items into those more relevant for teachers and researchers and those more specific for robot designers.

Their general comments on the robots' design indicated that they are not designed for group work: "What fails is that it seems to me…that in reality they are not meant to be used by more than a few children" - T1. They also admitted that working with robots is always associated with the robots falling from tables, "It will surely fall," stated T1 and T2 confirmed. They positively evaluated robots that can be easily lifted with one hand due to their size and form and imagined protecting materials that could be attached to the robot to absorb the impact.

A common consideration was the preliminary work with children that the robot requires. T1 mentioned that she always first explains what the children will face and what considerations they should have when manipulating the robot. Both agreed that it is essential to first go through embodied experiences related to spatial orientation, sequencing, and directionality. They also mentioned that more complex programming interfaces would benefit from unplugged activities related to challenging concepts, such as working on pattern recognition before using Ozobot's color codes.

We grouped general considerations and relevant aspects by the activity level and presented them in Figure 1.

4 DISCUSSION AND CONCLUSION

This paper addresses teachers' perspectives on commercial robots for preschoolers. By leading focus groups and lending the commercial robots to teachers to use with their pupils, we sought to understand the features the robots should have and other considerations related to classroom use with young children. As teachers play a crucial role in the successful implementation of educational tools, assessing their perspectives and experiences can shed light on the feasibility, usability, and pedagogical value of robots in the preschool classroom. By considering teachers' feedback, this study aims to inform future research and development efforts in designing robots better suited for educational settings. We aimed to contribute to understanding the advantages, challenges, and opportunities associated with using robots as educational tools in the preschool context. Earlier research has indicated that educators exhibit enthusiasm for educational robotics [20] and acknowledge its potential benefits. In our study, teachers were eager to try out some commercial robots they could use in their classrooms. They found that all the robots had the potential to be combined with preschool curricular content, and as vehicles to work on mathematics, spatial abilities, storytelling, and fun activities, such as robot races or making the robot carry messages between groups. However, previous research found that teachers generally hold unfavorable views regarding using robots within educational institutions which has been associated with the technical skills teachers should have to implement robotics curricula [20]. We believe that well-designed robotic kits should not require previous technical knowledge and be accessible to children and teachers, specifically in the context of kindergarten where CT could be taught in a simplistic and intuitive way. We consider that robots could be designed to support teachers instead of burdening them with the responsibility of learning how to use them, and we, as researchers, designers, and developers, should invest our efforts in creating user-friendly robots to be used in real-world contexts, such as educational settings. By doing so, we can alleviate the additional pressure placed on teachers, who already face the demands of an educational curriculum and extensive teaching responsibilities. Our study contributes to understanding how robots could seamlessly be integrated into kindergarten classrooms by contributing a set of design considerations to develop robots for this specific educational environment.

4.1 General Considerations for Developing Robots for Kindergarten

From our findings, we derived general considerations for developing robots to be used in kindergarten, useful for researchers, designers, and developers.

Design considerations for the development and design of a robotic kit (both robot and programming interface):

- ATTRACTIVE DESIGN AND INNOVATIVE INTERACTION. Robotic kits should be attractive and propose new modalities

of child-robot interaction that stimulate children's participation and creativity.
- COLOR CONSISTENCY BETWEEN INSTRUCTIONS AND ROBOT. It is important to maintain the same associations of colors throughout the activity.
- NEAREST INSTRUCTIONS AND ROBOT. The children could "get lost" if there are too many steps between the program and the robot's action. To easily follow program execution and support debugging, the program should be close to the robot.
- DIVERSITY OF INSTRUCTIONS. The robotic kit should allow the robot to interact with the user and environment in a variety of ways. It should support loops and conditionals. The instructions should be fun and interesting but also familiarize children with advanced programming syntax ("real coding").
- UPLOADING PROGRAMS SHOULD BE EASY AND INTUITIVE. Programming the robot should not require many steps and the programs should be easy to debug, upload and extend.

Design Considerations for the development and design of features specifically related to the robot:

- MULTIMODAL FEEDBACK. This feature would help reinforce the robot's actions and guide children in the activity. Teachers mentioned that multisensorial cues help to better understand the robot's actions. They also proposed that the robot could guide the activities by, for example, saying where the child has to start the activity and indicating errors and successes.
- ROBUST AND EASILY GRASPABLE WITH ONE HAND. Young children have little hands and are more prone to drop objects from their hands. A robot should be robust [9] because it may fall at some point in the activity. Being easy to grab with one hand could help prevent falls and ensure its durability. At this point, not only does the size matter, but the robot needs to have some affordance to grab it easily without slipping out. As the falling seems inevitable, it could have attached materials to absorb the impact.
- EASY BATTERY CHARGING, EXCHANGE, AND EXTENDED BATTERY DURATION. The battery should have a duration of 30-60 minutes to enable the robot's use in classes. Charging and exchanging batteries should be easy.
- PREDICTABLE AND PRECISE MOVEMENTS AND SHORT STEPS. The robot should not present unexpected behaviors, and its movements should be precise. Long steps and fast movements are potential limitations.
- LOW-COST FIXING AND MAINTENANCE. The robots should be easy to fix and the price of the components should be low.

Programming interface considerations are:

- ADEQUATE MATERIALITY, ACCESSIBILITY, AND AMOUNT OF ELEMENTS: Materials should have an adequate size and form, be made of durable materials, and be accessible for low vision and blind children. The quantity of coding elements should not restrict the programming of long paths.Also, materials should be easy to build and to be replaced.
- INTUITIVE INSTRUCTIONS. The teachers expect clear programming concepts represented in an intuitive way that does not require memorizing the instructions.
- VISIBILITY OF CODE. The instructions should enable users to visualize the program.
- PREVENTING SYNTAX ERRORS. The affordances of the coding elements should prevent syntax errors.

In the future, it would be crucial to incorporate some of the design considerations identified in our study and test them in real-world educational settings to support teachers and engage children in learning CT and bridge the gap between theoretical research and practical implementation. This would also be an opportunity to incorporate children's feedback, which is also crucial for the success of the activities.

5 LIMITATIONS

Access to the robots was a limiting factor of this study, as there are more robots with a tangible programming interface in the market that allow work with control structures. In our previous study, we identified 11 robots (see Table 3 [4]) with tangible user interfaces that allow working with control structures. Although we only worked with four robots, they represent all different types of conditionals and different manners of integration of conditionals with the main program identified in [4].

Another limitation was the size of our focus group. Working with only two teachers allowed us to maintain the same working group over an extended period of time and enrich the evaluations with insights about already evaluated robots that appeared in the following sessions. The teachers that we invited work with preschoolers and have broad experience in teaching computing, which allowed them to better visualize the possible implementation of the robots in classrooms and test them with children.

Another item to remark is that, as we conducted a focus group, not all aspects were discussed for all robots. For example, there were comments on the step size of Qobo and Botley but not on KIBO. With the relevant aspects identified in this work, we plan to conduct a comparative analysis of the four robots to provide a more in-depth evaluation of each robot.

ACKNOWLEDGMENTS

This work was supported by national funds through Agencia Nacional de Investigación e Innovación, Uruguay - FSED_2_2021_1_169697, PhD scholarship of Comisión Académica de Posgrado, Uruguay and Fundação para a Ciência e a Tecnologia, Portugal - project UIDB/50009/2020-, and by the Portuguese Recovery and Resilience Program (PRR), IAPMEI / ANI / FCT under Agenda C645022399-00000057 (eGamesLab). We thank the Computational Thinking department in Plan Ceibal, Uruguay for providing Botley and Ozobots used in the present study. We would like to express our sincere gratitude to the teachers that participated in the study.

REFERENCES

[1] Charoula Angeli and Nicos Valanides. 2020. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior* 105 (2020), 105954.

[2] World Medical Association et al. 2013. World Medical Association Declaration of Helsinki: ethical principles for medical research involving human subjects. *Jama* 310, 20 (2013), 2191–2194.

[3] Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, Maria Pascale, Camila Hergatacorzian, and Gonzalo Tejera. 2022. Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. In *Interaction design and children*. 422–429.

[4] Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, Gonzalo Tejera, Kerry Peterman, and Guillermo Trinidad. 2022. A Systematic Review of Technologies to Teach Control Structures in Preschool Education. *Frontiers in psychology* 13 (2022), 2840.

[5] Henry Barnes. 1980. An Introduction to Waldorf Education. *Teachers College Record* 81, 3 (1980), 1–9. https://doi.org/10.1177/016146818008100301 arXiv:https://doi.org/10.1177/016146818008100301

[6] Marina U Bers, Carina González-González, and Mª Belén Armas-Torres. 2019. Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education* 138 (2019), 130–145.

[7] Stefania Bocconi, Augusto Chioccariello, Panagiotis Kampylis, Valentina Dagienė, Patricia Wastiau, Katja Engelhardt, Jeffrey Earp, Milena Horvath, Eglė Jasutė, Chiara Malagoli, et al. 2022. Reviewing computational thinking in compulsory education: state of play and practices from computing education. (2022).

[8] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[9] Fernando Garcia-Sanjuan, Javier Jaen, Vicente Nacher, and Alejandro Catala. 2015. Design and Evaluation of a Tangible-Mediated Robot for Kindergarten Instruction. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology* (Iskandar, Malaysia) *(ACE '15)*. Association for Computing Machinery, New York, NY, USA, Article 3, 11 pages. https://doi.org/10.1145/2832932.2832952

[10] Kyriakoula Georgiou and Charoula Angeli. 2019. Developing preschool children's computational thinking with educational robotics: The role of cognitive differences and scaffolding. In *16th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA 2019*. 101–108. https://doi.org/10.33965/celda2019_201911l013

[11] Yen Air Caballero González and Ana García-Valcárcel Muñoz Repiso. 2018. A Robotics-Based Approach to Foster Programming Skills and Computational Thinking: Pilot Experience in the Classroom of Early Childhood Education. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality* (Salamanca, Spain) *(TEEM'18)*. Association for Computing Machinery, New York, NY, USA, 41–45. https://doi.org/10.1145/3284179.3284188

[12] Kay Yong Khoo. 2020. A Case Study on How Children Develop Computational Thinking Collaboratively with Robotics Toys. *International Journal of Educational Technology and Learning* 9, 1 (2020), 39–51.

[13] KinderLab Robotics. 2023. KIBO Robot. https://kinderlabrobotics.com/kibo/. https://kinderlabrobotics.com/kibo/ Accessed: 2023-06-01.

[14] Learning Resources. 2023. Botley Robot. https://botleybot.com/. https://botleybot.com/ Accessed: 2023-06-01.

[15] Kate I McCormick and Jacob A Hall. 2022. Computational thinking learning experiences, outcomes, and research in preschool settings: a scoping review of literature. *Education and Information Technologies* (2022), 1–36.

[16] Ana García Valcárcel Muñoz-Repiso and Yen Air Caballero-González. 2019. Robotics to develop computational thinking in early Childhood Education. *Comunicar* 27, 59 (2019), 63–72. https://doi.org/10.3916/C59-2019-06

[17] Ki Won Nam, Hye Jeong Kim, and Suyoun Lee. 2019. Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. *The Asia-Pacific Education Researcher* 28, 5 (2019), 387–397. https://doi.org/10.1007/s40299-019-00438-4

[18] Ozo EDU. 2023. Color Codes for Ozobot. https://static.ozobot.com/assets/5c9cf3b3-ozobot-color-chart-and-guide-2023.pdf. https://static.ozobot.com/assets/5c9cf3b3-ozobot-color-chart-and-guide-2023.pdf Accessed: 2023-06-01.

[19] Ozo EDU. 2023. Ozobot Robot. https://ozobot.com/. https://ozobot.com/ Accessed: 2023-06-01.

[20] Stamatios Papadakis, Julie Vaiopoulou, Eirini Sifaki, Dimitrios Stamovlasis, and Michail Kalogiannakis. 2021. Attitudes towards the use of educational robotics: Exploring pre-service and in-service early childhood teacher profiles. *Education Sciences* 11, 5 (2021), 204.

[21] Sarah Pila, Fashina Aladé, Kelly J. Sheehan, Alexis R. Lauricella, and Ellen A. Wartella. 2019. Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers & Education* 128 (2019), 52–62. https://doi.org/10.1016/j.compedu.2018.09.006

[22] Alex Pugnali, Amanda Sullivan, and Marina Bers. 2017. The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice* 16, 1 (2017), 171–193.

[23] Emily Relkin, Laura E de Ruiter, and Marina Umaschi Bers. 2021. Learning to code and the acquisition of computational thinking by young children. *Computers & education* 169 (2021), 104222.

[24] Robobloq. 2023. Qobo Robot. https://www.robobloq.com/product/Qobo. https://www.robobloq.com/product/Qobo Accessed: 2023-06-01.

[25] Evgenia Roussou and Maria Rangoussi. 2019. On the use of robotics for the development of computational thinking in kindergarten: educational intervention and evaluation. In *International Conference on Robotics and Education RiE 2017*. Springer, 31–44.

[26] Anika Saxena, Chung Kwan Lo, Khe Foon Hew, and Gary Ka Wai Wong. 2020. Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher* 29, 1 (2020), 55–66.

[27] X. Christine Wang, Youngae Choi, Keely Benson, Corinne Eggleston, and Deborah Weber. 2021. Teacher's Role in Fostering Preschoolers' Computational Thinking: An Exploratory Case Study. *Early Education and Development* 32, 1 (2021), 26–48. https://doi.org/10.1080/10409289.2020.1759012 arXiv:https://doi.org/10.1080/10409289.2020.1759012

[28] Jeanette Wing. 2011. Research notebook: Computational thinking—What and why. *The link magazine* 6 (2011), 20–23.

[29] Jeannette M Wing. 2006. Computational Thinking. *Commun. ACM* 24, 3 (2006), 33. https://doi.org/10.1145/1118178.1118215 arXiv:=

A RELEVANT ITEMS THAT CAN BE EVALUATED IN ROBOTS

Practical aspects identified:

- General aspects
  - Size (adequate size - too big/small)
  - Form (easy to lift with one hand - difficult to lift with one hand)
  - Fragility (robust - fragile)
  - Cost of fixing and maintaining (cheap to fix and maintain - expensive to fix and maintain)
- Battery
  - Battery charging (battery easy to charge - difficult to charge)
  - Battery duration (lasting long battery - battery goes empty fast)
- Movements
  - Step size (adequate step size - step too big/small)
  - Velocity of movements (moves with correct speed - too slow/fast)
  - Movements precision (precise - imprecise)
  - Unexpected behaviors (presents unexpected behaviors - do not resent unexpected behaviors)
- Coding elements
  - Coding elements size (adequate size of coding elements - coding elements too big/small)
  - Coding elements form (adequate form of coding elements - coding elements difficult to manipulate)
  - Amount of coding elements (sufficient amount of coding elements (blocks, cards) - limited amount of coding elements)
  - Precision of programming concepts (easy to understand commands - too abstract commands)
  - Color coherence between robot and code (colors are used to connect coding elements with the robot - there is no color relation between coding elements and the robot)
- Program uploading
  - Program uploading complexity (easy to upload the program - difficult to upload the program)
  - Uploaded program extension (easy to extend uploaded program - difficult/impossible to extended uploaded program)

Pedagogical aspects were:

- Available commands
  - Interaction with the user (it is possible to incorporate interaction with the users - no interaction with users supported)
  - Interaction with diverse sensors (offers possibility to work with diverse sensors - does not allowed to work with sensors)
  - Control over diverse actuators (allows to control diverse actuators - does not allowed to control actuators)
  - Loops support (allows to incorporate loops easily - does not support loops)
  - Conditionals support
    * Rigidity (flexibility in conditional statement building - conditionals are rigid)
    * Syntax (conditionals syntax easy to understand - complex syntax of conditionals)
    * Cognitive demand (conditional statement is evaluated one time - conditional statement is constantly evaluated)
  - Advanced programming (incorporates advanced programming syntax - does not incorporate advanced programming syntax)
- Coding elements
  - Commands complexity (coding elements are easy to understand - coding elements are abstract and must be learned)
  - Accessibility of coding elements (coding elements are accessible for users with low vision and blind - coding elements are not accessible)
  - Visibility of the program (programming interface makes the program visible - programming interface offers no visual support for the program)
  - Errors detection (coding elements help to detect programming errors - errors are first visible when the robot executes the program)
- Scaffolding
  - Previous work (does not require previous work - requires previous work)
  - Multimodal output (uses multimodal output (movements, lights, sounds) to communicate its actions - does not use multimodal output to communicate its actions)
  - Guidance during activity (guides the activity - does not guide the activity)
- General considerations
  - Connection with teaching curriculum (easy to connect with teaching curriculum - difficult to integrate with classroom activities)
  - Target age (adequate for preschoolers - targets older children)
  - Collaboration (prompts collaboration - designed for the individual use)
  - Program-robot distance (you can program the robot (almost) directly - programming the robot requires too many steps)

Motivational aspects identified were:

- Interaction with the child (can interact with the child - there is no child-robot interaction)
- Attractive design (has attractive design - is not attractive)
- Interesting commands (has fun and engaging commands - the commands are not very engaging)
- Innovation (allows to work in a way that is not possible with other robots - is similar to other robots)

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Appendix 6

Iterative Design and Empirical Evaluation of Conditionals for Robotito

Ewelina Bakala, Gonzalo Tejera, and Juan Pablo Hourcade. *An Iterative Design and Empirical Evaluation of Conditionals for Robotito.* Tech. rep. Udelar. FI., 2024. URL: https://hdl.handle.net/20.500.12008/45833

Author's contribution The author conceptualized the research idea, designed the methodology, and developed the data collection strategies. She conducted the evaluations with teachers, co-directed educational robotics activities and evaluations with children, and analyzed and visualized the data. She also led and actively participated in the writing process.

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Technical Report

An Iterative Design and Empirical Evaluation of Conditionals for Robotito

Ewelina Bakala, Gonzalo Tejera, Juan Pablo Hourcade

September 11, 2024

Contents

1 Introduction

Computational thinking (CT) is the process of formulating problems and solutions in a manner that allows a computer (whether human or machine) to effectively execute them [11]. This skill is considered essential for an active participation in the digital world [22], prompting its integration into educational curricula globally [22, 5].

Researchers emphasize the importance of CT learning at the early childhood stage, and empirical studies confirm that it is viable to teach CT concepts even at preschool level [16, 17, 9, 21]. Many of these studies use robots to teach CT, as they provide a concrete reference system for abstract problems.

Robotito is an educational robot developed as an open-source and open-hardware platform to introduce computational thinking to young children [27] (see Section 2 for more details about the robot). It has shown potential in stimulating CT development in preschoolers [8]. There is evidence that it is also well-suited to support groups of children learning together with one robot [1].

Robotito enables children to work on sequencing tasks that include abstraction, decomposition, route planning, and debugging. However, it consistently responds the same way to coding cards, which prevents it from addressing advanced programming concepts like conditionals. Conditionals are a critical component of many CT definitions [6, 10, 24], and validated CT tests for children aged 5 to 6 [18, 30] evaluate understanding of conditionals.

Recognizing the importance of conditionals, we decided to extend Robotito's capabilities to include this concept. We developed three prototypes for integrating conditionals into Robotito's functionality and a simulator to illustrate these ideas without needing to implement them in the physical robot. We evaluated our prototypes with experienced teachers to identify strengths and weaknesses of each prototype, and select the most appropriate for preschool context (see Section 3).

In the next step, we incorporated activities with conditionals and Robotito's simulator into Robotito curricula. The second part of this work (see Section 4) presents the results of an ER intervention with Robotito aimed to prompt the development of CT in level 5 kindergarten children. To measure the impact of the intervention, we adapted two validated CT tests and administered them before and after the activities. Additionally, the children's understanding of Robotito related concepts was assessed using a custom assignment, the Robotito Test.

2 Robotito

Robotito is an educational robot developed at Universidad de la República (Uruguay), designed to teach children CT concepts such as trajectory planning, sequencing, decomposition, and debugging.

On its underside, it has a sensor that allows to detect color cards placed on the floor. It responds to these cards by changing its movement direction according to the detected

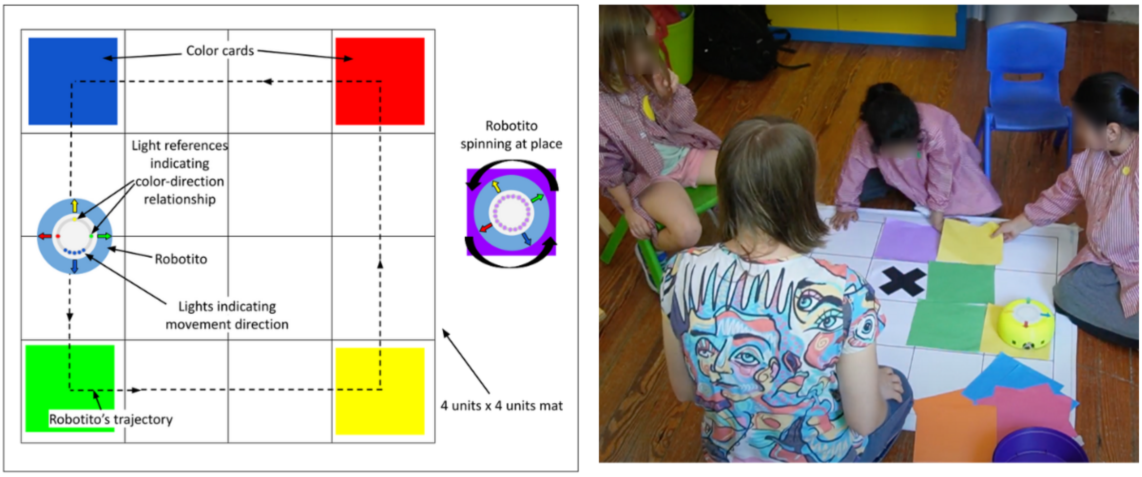


Figure 1: *Left: A schema of Robotito's response to color cards. The robot moves forward with yellow, left with red, backward with blue and right with green. Purple makes it spin. Right: An example of children solving a programming task.*

color: yellow makes it move forward[1], red to the left, blue makes it move backward, green to the right, while purple makes it spin (see Figure 1). It also responds with lights to the detected color cards. The robot indicates with four LEDs the direction it will move after sensing a particular color (see light references indicating color-direction relationship in Figure 1). When it detects a yellow, red, blue, or green card, additional LEDs light up to indicate the robot's movement direction. In the case of a purple card, all lights illuminate in purple.

During the ER actvities it is typically used with square color cards and a white mat divided into 4×4 units, 20cm×20cm each. The mat provides a homogeneous white background that enhances card detection by covering any distracting floor colors.

3 Development of conditionals

We evaluated different ideas for implementing conditionals in Robotito with three experienced teachers using prototypes with a different grade of fidelity.

3.1 Conditionals prototypes

We developed three ideas for implementing conditionals: color frame, musical mode, and split card. Each of these allows the robot to respond differently to the same coding card based on the evaluation of boolean expressions.

[1] Robotito has no front, so the relation to directions "forward", "backward", "left" and "right" are used only to distinguish its four predefined directions.

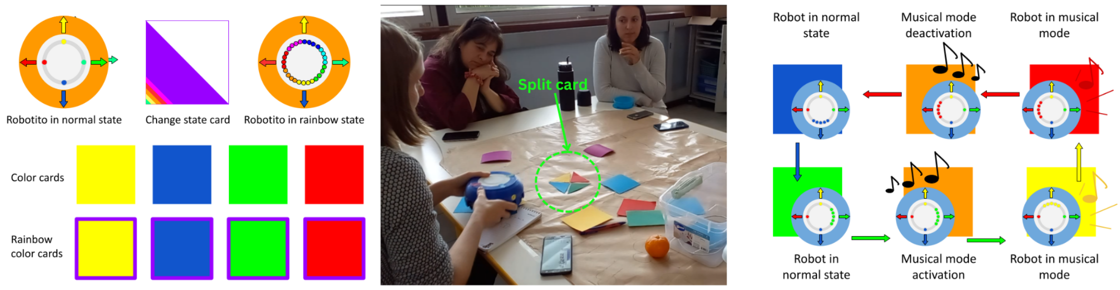


Figure 2: *On the left: An example of the implementation of color frame prototype. Depending on the state the robot responds to color cards or to rainbow color cards. In the middle: Paper prototype of the split card. When the robot approaches the card from the bottom, it senses blue and moves in the direction indicated by the blue arrow. When it approaches from the right, it senses green and moves in the direction indicated by the green arrow. Approaching from the left, it senses yellow, and from the top, it senses red. On the right: An example of musical mode activation and deactivation. The robot moves according to the color cards and produces activation/deactivation sound when passing over orange card and plays note associated with the color when passing over yellow and red card.*

3.1.1 Color frame

This prototype enables Robotito to respond differently to color cards based on the it's current state. In addition to the standard color cards, we introduce color cards with frames and a card that changes the robot's state (see Figure 2). In its normal state, Robotito detects color cards and ignores rainbow color cards[2]. In the rainbow state, Robotito does the opposite: it detects rainbow color cards and ignores color cards. The change state card is used to switch between states. Consequently, the robot behaves differently when encountering the same event (color card detection), depending on its state. The conditional logic expressed by this prototype is:

if $mode == rainbow$ **then**
  if $card == rainbowCard$ **then**
    $move(cardColor)$
  end if
else
  if $card == colorCard$ **then**
    $move(cardColor)$
  end if
end if

[2]The rainbow state is an example of a state the robot could implement. In our evaluations, we discussed other state changes, such as becoming angry or happy, to illustrate with a concrete example the new functionality.

3.1.2 Split card

The split card is a multicolor card containing four sections (see Figure 2). Each section can have a different color, although it is also possible to have sections with repeated colors. The direction in which the robot will move after detecting the card depends on the side from which the robot approaches it. This allows a single card to encode up to four different directions for the robot. The card can be rotated to solve a programming task. The conditional logic expressed by this card is:

if $comingFrom == bottom$ **then**
  $move(bottomCardColor)$
else if $comingFrom == right$ **then**
  $move(rightCardColor)$
else if $comingFrom == left$ **then**
  $move(leftCardColor)$
else
  $move(topCardColor)$
end if

3.1.3 Musical mode

This prototype incorporates two states for the robot: the normal state and the musical state. In the musical state, the robot executes its usual direction changes in response to color cards but also produces different sounds for each color card (see Figure 2). The state change occurs when the robot passes over the change state card (orange card) and is indicated by an activation or deactivation sound. This functionality allows the robot to perform an additional action—sound reproduction—when it is in musical mode. The conditional expressed by this prototype is:

if $mode == musical$ **then**
  $playSound(cardColor)$
end if

3.1.4 Robotito simulator

To evaluate our ideas without the need for immediate implementation in the physical robot, we developed Robotito simulator. It is a digital version of the robot that can be deployed as a desktop or Android application. It reflects the behavior of the robot and has been extended to incorporate additional features that we wanted to evaluate. The simulator was developed using the Processing [3], a programming language build on top of Java that facilitates rapid prototyping of interactive systems. The code of the musical mode and split card simulators can be found in the following repositories: https://github.com/ewelinka/robotitoMusical, https://github.com/ewelinka/robotitoSplitCard.

[3] https://processing.org/

Id	Date	Participants	Prototype evaluated	Evaluation type
1	07.06.23	P1, P2	Color frame Musical mode Split card	Focus group with both teachers based on oral explanation, paper prototype of new cards, and a researcher simulating Robotito's actions "by hand".
2	02.08.23	P1, P2	Musical mode Split card	Individual interview with each teacher using on screen Robotito simulator.
3	15.08.23	P3	Musical mode	Oral explanation of the new functionality and an interactive instance with the first implementation of musical mode in Robotito.
4	19.08.23	P3	Musical mode	Asynchronous feedback on a video generated using Robotito's simulator.

Table 1: *Summary of the evaluation sessions.*

.

During evaluations, we utilized the simulator as a desktop application and also used it to generate videos, which were later shared with teachers. Examples of simulations of the musical mode and split card can be found on YouTube[4].

3.2 Methodology

We conducted four evaluation activities involving three teachers. Each activity was video recorded and analyzed.

3.2.1 Participants

The participants included: a computing teacher who works with preschoolers and early primary school students at a private educational center (P1); a teacher who teaches at both public and private institutions at the preschool and primary school levels (P2); and a preschool teacher from a public institution (P3). P1 and P2 participated in the first two evaluation sessions, while P3 participated in the third and fourth evaluation. All participants work in schools in Montevideo, Uruguay and had more than 15 years of teaching experience.

3.2.2 Evaluation activities

Before evaluating the prototypes, all teachers were familiarized with Robotito and its responses to color cards. During the evaluations, we presented our ideas using various strategies: oral explanations, paper prototypes of new color cards, simulating Robotito's actions "by hand" with the robot turned off, utilizing the Robotito simulator, and using Robotito itself. As we made improvements between the evaluations, each session employed different evaluation materials. Details of each session are provided in Table 1.

[4] https://www.youtube.com/playlist?list=PL575oRsFVM9qjtbGgaP6RRVXiNXM176Wu

3.3 Results

Each evaluation provided valuable input, helping us identify the most appreciated ideas and focus on potential improvements for subsequent evaluations.

3.3.1 Evaluation #1 (low fidelity prototypes)

We conducted a focus group with P1 and P2 to discuss all the ideas. The session involved a verbal explanation of each prototype, paper prototypes of new cards, and a researcher simulating Robotito's actions "by hand".

Color frame The idea of additional cards and an internal state that completely alters the robot's reaction to color cards was considered complex. P2 remarked that "everything changes" and that "there are two parallel universes."

To provide a concrete example that would justify the use of color frames, P2 suggested an activity based on missions. In this scenario, the robot might fall or crash if it follows a path composed of only one type of card (either directional cards or directional rainbow cards, depending on the state). The challenge would be to change the robot's state at the appropriate moment to avoid these unwanted situations and achieve the goal, although P2 admitted, "Still, it is difficult." P1 envisioned that borders could have textures influencing the robot's behavior, such as a green border with a grass texture that slows the robot down. She suggested that patterns on the borders could be more concrete and have an immediate effect that is easily understood at the preschool level.

Split card P1 questioned the benefits and new challenges introduced by the split card: "In the end, it does the same as if you put this (blue card) here. Why do you divide it into four? What makes it different?" P2 suggested that "Perhaps you can give the card with four colors to the child and the child has to decide how to rotate it." They noted that the robot's starting position and orientation should be predefined so the child can solve the task. P2 observed, "I have to consider where it starts, what will be the first color that it senses, and which direction it will go." Additionally, they mentioned the need to account for the number of coding cards or obstacles to ensure the children would use the split card. Both teachers admitted that using the split card was not straightforward. P2 stated, "You have to think well about the tasks," and P1 added, "You have to consider the context." P2 summarized these considerations by stating, "The use of the (split) card is somehow forced."

Despite these concerns, P2 remarked that she really liked this idea and P1 found it attractive.

Musical mode After the researcher's explanation, the teachers immediately focused on the idea of creating melodies using color cards. We discussed potential issues, such as when the robot should repeat two notes, causing it to loop between two cards, and then play a new note. In this scenario, the card with the new note should be placed while the robot is moving to interrupt the loop. P1 suggested that when the robot is in musical mode, it could maintain a fixed movement direction and only read the notes: "Do not modify the direction variable, leave it fixed." She was enthusiastic about working on sequencing using popular melodies or songs, such as "Baby Shark." She stated, "The song guides the order," and "the ear corrects you."

Summary All the prototypes presented challenges that were noted by the teachers. They proposed various ideas to overcome these challenges and considered classroom activities that could employ each prototype.

The color frame prototype was the only one that did not receive positive feedback, and the potential activities with it were considered difficult. Therefore, we focused further evaluations on the split card and musical mode.

3.3.2 Evaluation #2 (simulator)

During the second evaluation, we used the simulator to provide an interactive experience with the musical mode and split card prototypes. We worked with P1 and P2 separately.

Musical mode In the digital version of the musical mode, the card used to activate and deactivate musical mode would make a sound corresponding to on or off, with musical mode adding a musical note to the change of direction. This implementation of the musical mode was easily understood by both teachers. P1 remarked, "Ah! It [the color card] converts into notes. In addition to direction, it is a note." She also commented on the clarity of the activation and deactivation sound: "It is understandable that the sound is turning on and off. It's very clear." She was confident that children would understand the musical mode: "The only thing that changes is that it incorporates sound. It is not a substantial change. [...] It doesn't confuse, doesn't dazzle, this is what I'm saying."

Both teachers agreed that activating and deactivating the musical mode with the same card was a good idea. P1 noted, "with the same [card] it's easier, thinking in practice."

P1 envisioned composing simple melodies as a sequencing exercise. Although only four notes can be used with four colors, she did not see this as a problem: "At the initial level we do simple things," and "more sequence is more abstraction and more difficulty." She saw composing as engaging and emphasized that "when they [the children] are motivated they will want to spend more time working with the code."

P2 also showed interest in programming simple songs but saw working with only four sounds as a limitation. She revisited the idea from the first evaluation session, where the robot in musical mode does not change its direction but only reads the color cards as notes. This way, the robot would move as usual with color cards or go in a straight line while reproducing the sounds associated with the color cards. She considered combining these two modes challenging: "We are dividing [children's] attention between two different things, and we are working with young children." However, she saw it as viable for level 5 kindergarten (children aged 5 to 6) after some initial work with the concepts. The basic version of the prototype, in which the robot moves with the colors and in musical mode also reproduces the sound, was considered easy to understand: "The only thing that you add is that it makes sound, the movements do not change" (P2). An exercise where the children have to activate and deactivate the sound to create silence or sound in specific parts of the route was considered viable: "It's a good proposal," stated P2.

Split card The split card prototype was less discussed. P1 considered it accessible for the children since they only need to apply the color-direction rule that they already know. She imagined an introductory exercise where the normal color card is replaced by the split card to demonstrate that "it is the same."

P2 was more enthusiastic, stating, "It's incredible, I love it." She found it suitable for preschoolers and highlighted that it allows working on problems using a trial-and-error strategy.

Prototypes' ranking We asked the teachers to rank the prototypes based on their suitability for preschool-level education.

P1 found both ideas attractive for preschoolers. Regarding which prototype would allow her to offer more engaging activities, her preference was musical mode. While she found the split card idea attractive, she noted its limitation, stating, "It is not more than changing the direction."

P2 shared the same preferences, stating, "I would start with the musical one; I really like the musical one. Then the split card; I really like that one too."

Robotito simulator Both teachers spontaneously mentioned that the Robotito simulator could be used to introduce Robotito to children before working with the real robot. P1 commented on the benefit of having both virtual and tangible formats: "If I have to work on it [an activity with Robotito], I would like to explain a little bit of theory, [...] let them see it first [on the screen], and then we do it [with the robot]. This way, their anxiety decreases since they have already seen it and know how it works, and the child is more self-regulated." P2 found that "it [the simulator] is excellent to work beforehand" and "[in the simulator] we observe what it does and then, we translate it into [robot's] trajectory."

When asked if the digital version was understandable or perhaps too abstract, both agreed that the representation of the robot and the mat were appropriate. P1 remarked, "No, it is perfect, less is more."

Summary Both teachers ranked musical mode as their preferred prototype for young children due to its potential for developing playful activities involving sound and movement. It was deemed more engaging compared to solely focusing on directional changes, as seen with the split card.

Interestingly, both teachers highlighted the benefits of using the Robotito simulator to introduce Robotito to children before using the physical robot. They appreciated the combined use of virtual and tangible formats to provide better learning experience.

3.3.3 Evaluation #3 (high fidelity prototype)

During the third evaluation, the researcher that led the session presented to P3 the in-robot implementation of musical mode. The educator spontaneously began interacting with the robot, attempting to make it play the first part of Beethoven's composition "Für Elise." The challenge of composing a melody with four notes while the robot changes direction with each color was considered complicated. "It is too much," stated P3. She acknowledged that "they [the children] will love it" and that it "sparks creativity," but felt it was too complex for kindergarten.

The researcher mentioned that in previous evaluations, the idea of fixing the movement direction in musical mode had been proposed so that children could focus on the notes without needing to think about direction changes. However, this idea did not convince P3. She found using the same cards for both directions and music confusing. She explained that the idea could work "if these cards had a drawing of a musical note or something to differentiate them from the others, otherwise, it's a mess."

P3 considered the concept of passing in silence or making sound in some parts of the robot's route much more viable: "[The option] to go with sound or without sound is great." She was undecided on whether there should be two separate cards to activate and deactivate the musical mode, or if one card would suffice.

Summary During the third evaluation, the teacher found the musical mode engaging but too complex if the objective is composing melodies while managing direction changes. The idea of fixing the robot's direction in musical mode to focus on notes was not convincing. The concept of the robot passing in silence or making sound at specific points was seen as more viable. There was uncertainty about whether one or two cards should be used to activate and deactivate the musical mode.

3.3.4 Evaluation #4 (simulator's video)

The video of the musical mode simulator[5] presented to P3 was deemed "super clear." She found that it accurately reflects Robotito's behavior that she experienced in the previous evaluation session: "I think I was watching exactly how the robot works." However, she noted the absence of visual feedback to indicate whether Robotito is in musical mode: "What caught my attention [...] is that there's nothing visually indicating that it's in musical mode. I didn't see any different light turning on; you can only tell if you hear the sound or not."

[5]https://www.youtube.com/watch?v=Ju89amk-yTs

3.4 Discussion

3.4.1 Prototypes

All the prototypes we evaluated presented certain challenges, and exchanges with the teachers helped us focus on the most viable ideas and incorporate improvements.

The color frame prototype was considered too demanding, as it relied on an if-then-else condition that would require children to manage "two parallel universes," each with a different set of coding cards. In contrast, the other prototypes were simpler, requiring only one additional coding card. Both were found attractive, but the decisive factor in the teachers' preferences was the potential to explore new actions in musical mode. Sound reproduction was considered engaging, and activities involving passing through certain parts of the robot's route in silence or with sound were seen as accessible for preschoolers. While the teachers were enthusiastic about composing melodies, they were also aware of the difficulties associated with combining sound and direction or separating the musical mode from the directional mode. The viability of music composition should be validated in future studies.

The interaction with the teachers helped validate specific aspects of the musical mode implementation. For example, the teachers suggested using a single card to activate and deactivate the mode. They also provided examples of activities and identified potential improvements, such as adding visual cues to indicate whether the robot is in musical mode.

3.4.2 Robotito simulator

The teachers saw the simulator as a valuable tool for introducing Robotito to children. They noted that presenting the robot on a screen or projecting it onto a whiteboard would be less distracting for the children since it cannot be touched or grabbed. They envisioned using it to explain the color-direction relationship or introduce a specific activity before interacting with the real robot, thereby reducing the children's anxiety.

The enthusiastic reaction of the teachers helped researchers envision the simulator's use as a tool not only for introducing the robot but also for practicing programming individually. The same code used to generate the desktop simulation can be deployed as an Android application, allowing interaction with digital coding cards and an on-screen Robotito by dragging them with a finger. Given that each public preschool in Uruguay is equipped with Android tablets, we began considering incorporating programming Robotito on the tablet as part of Robotito's curriculum.

Although the simulator can help in practicing robot programming individually, it is essential to combine its use with hands-on experiences with the actual robot. While the simulated Robotito scenario is useful for practicing trajectory programming, it lacks the ability to incorporate new elements, making it challenging to engage children through activities like personalizing the robot or adding characters and decorations to build a narrative. The limited flexibility of the simulated scenario also hampers integration with preschool curricula. We consider that simulators can reinforce the learning experience, but they should be complemented by tangible robots that offer concrete materials and greater flexibility for incorporating new elements.

4 Field study

Drawing from the insights gained during prototypes evaluation, we enhanced Robotito's capabilities by adding a musical mode. We then incorporated activities involving both Robotito's simulator and the new musical mode into the curriculum implemented during the field study.

The following research questions guided the field study:

- What was the impact of activities with Robotito on children's computational thinking development?
- Can preschool children understand and successfully use conditionals?
- Is there a correlation between the validated PC tests and between the tests and Robotito Test?

4.1 Methodology

We conducted a study with a quasi-experimental design to evaluate whether a curriculum of eight educational robotics (ER) sessions with Robotito influenced the development of computational thinking in the active group, compared to the control group. The study involved an active group (AG) composed of two classes (A1, A2) and a control group (CG) consisting of one class. The activities focused on trajectories planning, sequencing, and conditionals. We adapted two validated CT tests for preschoolers to measure CT levels before and after the activities. We also developed a custom assignment, the Robotito Test, to evaluate children's understanding of the concepts introduced in the ER sessions. This test was administered after ER sessions and only to the AG.

4.1.1 Robotito

As discussed in Section 2, Robotito moves in response to five color cards: yellow, green, blue, red and purple. In this study, two additional color cards were introduced: orange, for conditional music reproduction, and pink, to familirize children with the concept of modularization. After detecting the pink card, Robotito moves one step forward (yellow direction), then one step right (green direction), and stops the movements. During the execution of this "pink step," the robot ignores all other color cards.

4.1.2 Participants

We worked with 56 preschoolers from level 5 (5 to 6 years old) at a public kindergarten in Montevideo, Uruguay. Two classes (A1 with 17 students and A2 with 19 students) formed the AG, and one class (20 students) was the CG. A1 attended the kindergarten in the morning, and A2 and CG were the afternoon groups. The assignment to the AG and CG was decided by the kindergarten's principal.

4.1.3 Activities

We conducted eight ER activities with Robotito between November and December of 2023 (see Table 2). The activities were designed taking into account lessons learned from two exploratory studies [3], new capabilities of the robot, and ideas from a preschool teacher who worked with Robotito in her classroom. A detailed description of all activities can be found in Appendix E.

4.1.4 Data collection and analysis

We assessed children's CT levels at two time points: before starting the activities with Robotito and at the end of all sessions. We also examined children's knowledge related to the concepts addressed in the activities with Robotito using the Robotito Test that we developed. All the tests were administered individually to the children, and three evaluators participated in the evaluation process. We performed statistical analysis and data visualization using the R programming language [6].

Additionally, we video recorded all the activities and the Robotito Test administration to enrich our analysis with qualitative data.

Evaluation Instruments We adapted two CT tests [18, 30] to measure CT before and after the activities. We chose them because they are the only existing validated tools for our age group. Additionally, we developed a custom test, the Robotito Test, to evaluate children's understanding of the concepts introduced during the ER activities.

The Beginners Computational Thinking Test (BCTt) [30] targets children aged 5 to 12 and consists of 25 items assessing Sequences (6), Simple Loops (5), Nested Loops (7), If-then Conditional (2), If-then-else Conditional (2), and While Conditional (3). We tailored the test by retaining items relevant to our study and reducing the number of questions related to Loops, as they were not explicitly addressed in our curriculum. Our adapted version comprises 13 items, covering Sequences (items 1 to 6), If-then Conditional (items 7 and 8), Simple Loops (item 9), and Nested Loops (items 10 to 13). These items correspond to items 1, 2, 3, 4, 5, 6 (Sequences), 19, 20 (If-then Conditional), 8 (Simple Loops), 13, 14, 15, 17 (Nested Loops) in the original BCTt. The tailored test used in the study can be found in Appendix A.

[6] https://www.r-project.org/

Activity	Date, class (nr of children)	Modality of work	Main goal of the session
#1	06.11.23 A1 (14) 06.11.23 A2 (18)	Whole class together.	To introduce Robotito and how it moves with yellow, red, green and blue color cards.
#2	10.11.23 A1 (16) 10.11.23 A2 (18)	The class splitted in two groups.	To reinforce how the robot responds to color cards through an embodied experience. To observe that the color cards should be placed in the robot's trajectory.
#3	20.11.23 A1 (16) 17.11.23 A2 (18)	The class splitted in two groups.	To understand that directing the robot depends on the color of the coding card and the robot's rotation. To reinforce that the color cards should be placed in the robot's trajectory.
#4	22.11.23 A1 (16) 20.11.23 A2 (19)	Whole class together.	To reinforce how the robot responds to color cards through more individual interaction with Robotito's simulator. To practice route planning, sequencing and sequence decomposition.
#5	24.11.23 A1 (15) 24.11.23 A2 (15)	The class splitted in two groups.	To plan Robotito's trajectories, select the corresponding color cards, and place it in space. To introduce conditional music reproduction using the orange card.
#6	27.11.23 A1 (14) 24.11.23 A2 (16)	The class splitted into small groups.	To practice coding Robotito's routes and reinforce how it responds to the orange card.
#7	29.11.23 A1 (14) 29.11.23 A2 (17)	The class splitted in two groups.	To introduce a pink card that makes the robot execute a prerecorded sequence of movements and stop.
#8	04.12.23 A1 (14) 04.12.23 A2 (18)	The class splitted into small groups.	To reinforce how the robot responds to the pink card and practice combining it with the other coding cards.

Table 2: *Summary of ER sessions.*

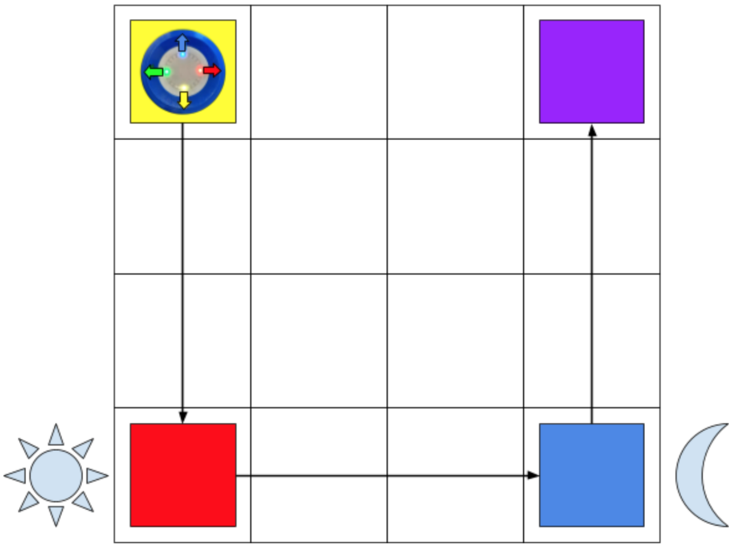


Figure 3: *Task 3 from Robotito Test. Children have to place orange cards so that Robotito makes sounds near the sun and is silent near the moon.*

TechCheck-K [18] is a kindergarten version of TechCheck (a validated CT test that targets children aged 6 to 9 [19]). It targets children aged 5 to 6, and evaluates 15 items related to Hardware/software (2 items), Debugging (2), Algorithms (5), Modularity (2), Representation (2), and Control Structures (2). We used a tailored version of the test that contained only items relevant in our study, resulting in 9 items focusing on Algorithms (items 1 to 5), Representation (items 6 and 7), and Control Structures (items 8 and 9). These items correspond to items 5, 8, 9, 10, 11 (Algorithms), 12, 13 (Representation), 14, 15 (Control Structures) in the original TechCheck-K. The tailored test used in the study can be found in Appendix B.

Robotito Test was developed to assess children's understanding of the concepts that we addressed in the ER activities. All the tasks represent on paper a typical activity setting (Robotito, a 4×4 units grid, and color cards). See Appendix D to visualize the test.

In the first task, children have to choose the colors of two coding cards to guide the robot to the purple card. The route and cards' locations are predetermined, requiring only the selection of cards' colors. This task assesses whether children understood the color-direction relationship and can deduce it from a particular robot's orientation to solve a specific sequencing task.

The second task entails designing a sequence of movements for the robot to reach the purple card while defining both the location and color of the coding cards.

In the third task, children's comprehension of conditional music reproduction is evaluated. Here, the U-shaped sequence of the robot's movements is predetermined, with corresponding cards already in place (see Figure 3). Children are tasked with positioning orange cards to activate or deactivate Robotito's sounds at two specific points along its route.

The fourth task evaluates the correct usage of the pink card. With the robot initiating its movements on the pink card and aiming to reach the purple card, children are asked to place the missing cards to complete the programming sequence.

We provided the children with four color cards (yellow, red, green, blue) to solve the task 2 and 4, and with two orange cards to solve the task 3. Each task was evaluated as correctly solved (1 point) or incorrectly solved (0 points) without decimal scores. Administration was conducted by a single researcher, who could scaffold children with questions like:

- "Can you show me with your finger how the robot will move?" (Task 2 and 4)
- "How does the robot move with the pink card?" (Task 4)

and explain issues that were different between the on-paper robotic task and the real robot acting:

- "It is ok to cover the robot with a color card, it's the same as placing it below the robot." (Task 2)
- "You can not cover printed color cards with the orange card." (Task 3)

These measures aimed to enhance qualitative analysis by providing insight into children's reasoning and comprehension, and ensure that the children understood the on-paper programming setting.

Statistical analysis We conducted statistical analysis to identify differences between PRE and POST scores across the AG (comprising A1 and A2) and CG in CT tests (tailored TechCheck-K and BCTt), correlations between CT tests, and correlations between Robotito Test and CT tests. We used the Shapiro-Wilk test [23] to assess the normal distribution of the samples and Bartlett's test [4] to examine the variance between groups. Group comparisons were performed using the two-sample t-test [7], the Wilcoxon rank-sum test [29], and factorial ANOVA [28]. Comparisons between PRE and POST results within the same group were made using the paired t-test [13] and the Wilcoxon signed-rank test for paired data [29]. Wilcoxon tests were used when the data did not follow a normal distribution. The correlation between the tests' scores was calculated using the Pearson correlation coefficient (PCC) [12] for data that follows a normal distribution, and the Spearman rank correlation coefficient (rho) [25] otherwise.

4.2 Results

We analyzed whether there was a difference between the AG (composed of A1 and A2 classes) and CG in overall scores (PRE and POST scores together) and in PRE and POST scores separately. We were also interested in determining whether there was

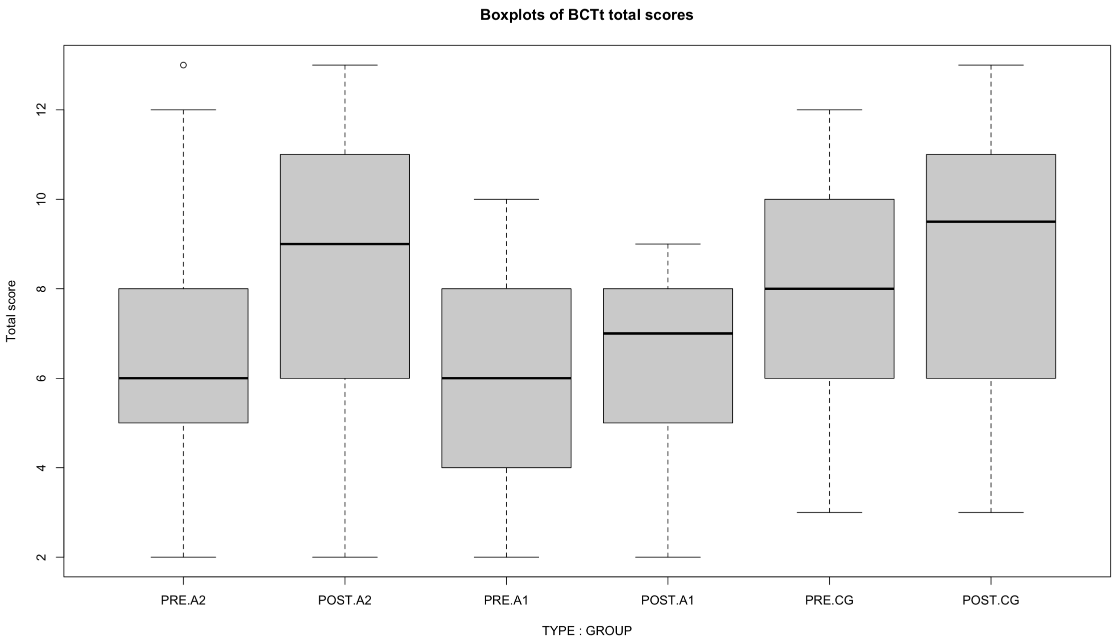


Figure 4: *Boxplots of PRE and POST total scores of each class.*

an improvement in any specific CT concept (Sequencing, If-then Conditional, Simple Loops, and Nested Loops in case of BCTt; Algorithms, Representation, and Control Structures in case of TechCheck-K) and if our results were similar to those reported in the literature. Finally, we analyzed if the tests' results show any correlation.

The tests' results can be consulted in Appendix G (BCTt), Appendix H (TechCheck-K), and Appendix I (Robotito Test) or downloaded from `https://drive.google.com/drive/folders/1FrFkTNljsb23nL3QeeY-3FdFHHOhwJ_v?usp=sharing`.

4.2.1 BCTt

PRE and POST scores in AG and CG All the samples that we compared had a normal distribution, so we performed two-sample and paired t-tests, and factorial ANOVA analyses.

Was there a significant difference in PRE or POST scores between AG and CG?

The two-sample t-test revealed a significant difference (p=0.018) between the BCTt scores of the AG (mean=6.889) and the CG (mean=8.2). Although the AG had a lower mean PRE score than the CG (see Table 3), this difference was not statistically significant (p=0.058). Both groups improved their performance in the POST test; however, the mean POST score of the AG (mean=7.417) did not manage to reach the mean PRE score of the CG (mean=7.75). There was no statistically significant difference between the POST scores.

We further analyzed PRE and POST scores by dividing the AG into classes (see

	PRE	POST
CG	7.75	8.65
AG	6.361	7.417
A1	6.0	6.053
A2	6.684	8.211

Table 3: *BCTt mean PRE and POST score of the AG, A1, A2 and CG.*

Table 3 and Figure 4) to identify any differences in performance between the active groups (A1 and A2) and between the active groups and the CG.

We found a significant difference in PRE test scores between A1 and CG (p=0.043), but not between A2 and CG, nor between A1 and A2. Similar results were observed when comparing POST scores: there was a significant difference only between A1 and CG (p=0.015).

Have the groups improved their scores?

We conducted paired t-tests separately for each group to assess whether there was a significant improvement within each group.

The paired t-test demonstrated that the AG significantly improved its BCTt score (p=0.034, mean difference=0.9), while the CG showed no significant improvement. Class-by-class analysis revealed that only A2 significantly improved its BCTt score (p=0.042, mean difference=1.526), whereas A1 showed no statistically significant improvement.

Does the change in scores over time depend on the group variable?

Factorial ANOVA confirmed a significant difference (p=0.019) between the AG and the CG overall scores (PRE and POST scores together), indicating that, in general, the CG performed better than the AG. At the interaction level, which considers both time and group, the only significant difference found was between CG:POST and AG:PRE (p-value=0.020).

The analysis of the three classes (A1, A2, CG) showed a significant difference (p=0.009) in scores only between A1 and CG groups. The interaction between PRE and POST conditions and the three groups indicated a significant difference only when comparing A1:PRE and CG:POST. The significant difference between A2:PRE and A2:POST scores identified using the t-test was not confirmed by the ANOVA analysis.

PRE and POST concept scores in AG and CG We grouped the scores of questions related to Sequencing, If-then Conditional, Simple Loops, and Nested Loops (see Section 4.1.4 for questions related to each concept) and compared the scores.

Which concepts related to CT improved over time?

To determine which concept scores improved over time, we compared PRE and POST scores of each group (AG, CG, A1, A2). In only four cases (A2 Sequencing, A1 If-then Conditional, CG and A2 Nested Loops) did the difference between PRE

Id in adapted BCTt used in our study	BCTt item id	Percentage of correct answers for the first educational stage in [30]	Percentage of correct answers in our study
1	1	93%	62%
2	2	93%	64%
3	3	91%	73%
4	4	89%	61%
5	5	76%	52%
6	6	87%	57%
7	19	71%	50%
8	20	41%	29%
9	8	77%	50%
10	13	76%	64%
11	14	55%	27%
12	15	27%	25%
13	17	96%	71%

Table 4: *Percentage of correct answers for items evaluated in our study. Comparison of Table 17 values from [30] and our score calculated as a mean PRE value of all groups (A1, A2, CG).*

and POST scores show a normal distribution. For these cases, we performed a paired t-test; for the others, we used a Wilcoxon signed-rank test for paired data.

Sequences and Nested Loops scores showed no significant difference when comparing PRE and POST scores of any group. The If-then Conditional score improved only in the CG (p-value=0.044), and Simple Loop score in the AG (p-value=0.022).

Comparison with other studies To determine if our results align with those reported in the literature, we analyzed studies that report BCTt scores. We found only two relevant studies providing a percentage of correct answers for our age group. The study by Zapata-Cáceres et al. [30] presents the percentage of correct answers by concept (see the first column in Table 5) and by individual question (see the third column in Table 4) for children aged 5 to 8. The second study by Zapata-Cáceres and Fanchamps [31] presents the percentage of correct answers grouped by concept specifically for 5-year-olds (see the second column in Table 5).

Percentage of correct answers of each item

Correlation analysis of the percentage of correct answers for each item revealed strong (rho=0.930) correlation between Zapata-Cáceres et al. [30] results and the PRE test results for all samples (AG and CG combined). Although the scores of [30] are consistently higher than ours (see Table 4), this discrepancy is expected, as they report scores for the first educational stage that includes children between 5 and 8, while our study was conducted with younger children aged 5 to 6.

Percentage of correct answers grouped by concept

When analyzing percentage of correct answers grouped by concept (see Table 5),

	Zapata et al. [30]	Zapata-Cáceres and Fanchamps [31]	Our study
Sequences	90%	59%	62%
If-then Conditional	44%	30%	39%

Table 5: *Percentage of correct answers for Sequences and If-then Conditional. Scores reported in Figure 15 in [30], Table 3 in [31], and our score calculated as a mean PRE value of all groups (A1, A2, CG).*

we found that our PRE scores for the entire sample (AG and CG combined) were similar to the results reported by Zapata-Cáceres et al. [30] and Zapata-Cáceres and Fanchamps [31], except for the Sequences scores from [30], which were significantly higher than our results.

Summary of BCTt results Our analysis showed that in general, CG performed significantly better than AG. However, when comparing only PRE or POST scores individually, no statistically significant difference was observed between the two groups. Between classes comparisons showed a significant difference between A1 and CG.

Comparison of PRE and POST scores of the groups showed that AG significantly improved its scores, and class-by-class analysis showed improvement only in the case of A2. When analyzing CT concepts, we observed only two improvements- the If-then Conditional score in CG and the Simple Loop score in AG.

The ANOVA interaction between time and group variables demonstrated a significant difference only when comparing A1:PRE and CG:POST, which indicates that the changes in scores over time (PRE, POST) do not depend on the group variable (AG, CG, or A1, A2, CG).

The BCTt percentage of correct answers for each item showed strong correlation between our PRE test results and those reported in [30]. When grouped by concept, our PRE results were similar to those in [30] and [31], except for the Sequences score reported in [30], which was higher than our result.

4.2.2 TechCheck-K

We analyzed if there was a difference between the AG and the CG in overall scores (PRE and POST combined) and between PRE and POST scores of each group (AG, CG, A1, A2). We were also interested if there was an improvement in any specific concept related to CT (Algorithms, Representation, and Control Structures).

PRE and POST scores in AG and CG We analyzed the distribution of the overall scores of the AG and the CG, and PRE and POST scores of all groups (AG, CG, A1, A2). Only overall AG scores and AG:PRE scores did not follow a normal distribution.

	PRE	POST
CG	4.65	5.0
AG	4.639	5.278
A1	4.418	5.176
A2	4.842	5.368

Table 6: *TechCheck-K mean PRE and POST score of the AG, A1, A2, and CG.*

Was there a significant difference in PRE or POST test between AG and CG?

To compare the scores, we used a two-sample t-test or, if the data had a non-normal distribution, we used the Wilcoxon rank-sum test. We observed that general TechCheck-K scores and PRE and POST scores were similar in AG and CG (see Table 6), and a statistical analysis confirmed that there was no significant difference between the scores.

We compared PRE and POST scores between active classes (see Table 6), and PRE and POST scores of each class compared to CG, but these more detailed analyses did not reveal any significant differences either.

Have the groups improved their scores?

We compared PRE and POST values of all groups to see if there were significant changes. In the case of A1, the difference between POST and PRE scores did not have a normal distribution, so we used the Wilcoxon signed-rank test for paired data to compare the data, in all other cases, we used the paired t-test. None of the comparisons showed a significant difference between the scores.

PRE and POST concept scores in AG and CG We grouped the scores of questions related to Algorithms, Representation, and Control Structures (see Section 4.1.4 for questions related to each concept) and compared the scores.

Which concepts related to CT improved over time?

To define which concepts improved over time, we compared PRE and POST scores of each group (AG, CG, A1, A2). We used the paired t-test, and in the cases of score difference with non-normal distribution (A2 Algorithms, AG and A2 Representation, and all groups in Control Structures), we used the Wilcoxon signed-rank test for paired data. Only the AG Control Structures significantly improved over time.

Comparison with other studies We identified three publications that report scores of individual items [18] or CT concepts [20, 14] at kindergarten level. The studies involved 89 kindergarten students in [18], 395 in [20], and 24 in [14].

Percentage of correct answers of each item

Relkin and Bers [18] report the percentage of correct answers of each TechCheck-K item (see the third column in Table 7). We observed a strong correlation (rho=0.782) between the percentages of correct answers of the entire group (AG and CG combined)

Id in adapted TechCheck-K used in our study	TechCheck-K item id	Percentage of correct answers for kindergarten estimated from [18]	Percentage of correct answers in our study
1	5	58%	48%
2	8	59%	68%
3	9	60%	57%
4	10	55%	39%
5	11	33%	27%
6	12	33%	27%
7	13	31%	41%
8	14	67%	84%
9	15	56%	73%

Table 7: *Percentage of correct answers for items evaluated in our study. Comparison of values estimated from Figure 4 from [18] and our scores calculated as a mean PRE value of all groups (A1, A2, CG).*

in the PRE evaluation and those reported in their study (see Table 7).

Percentage of correct answers grouped by concept

Relkin et al.'s [20] study provides a percentage of correct answers for each CT concept (see the first column in Table 8). Our results are similar, with our scores being slightly higher for Algorithms and Representation and considerably higher for Control Structures. Lin et al.'s results [14] (see the second column in Table 8) were slightly higher than our results.

Summary of TechCheck-K results We compared the overall, PRE, and POST scores of AG and CG and found no significant differences. There was also no difference when we compared the scores across classes.

None of the groups significantly improved its general, PRE, or POST score. Only AG improved the score of Control Structures over time.

As the comparisons of the PRE and POST scores showed no differences, we did not conduct ANOVA to analyze if the change in scores over time depends on the group variable, as we did when analyzing BCTt scores.

We observed a strong correlation between the percentage of correct answers of each item for our entire sample and those reported in Relkin and Bers [18]. When analyzing percentage of correct answers by concept, our results were similar to those in Relkin et al. [20], although our score for Control Structures was considerably higher. The scores reported by Lin et al. [14] were slightly higher than our results.

4.2.3 Robotito Test

We conducted a quantitative analysis of the scores and a qualitative analysis of the videos from the test administration process.

	Relkin et al. [20]	Lin et al. [14]	Our study
Algorithms	42%	53%	48%
Representation	32%	34%	34%
Control Structures	58%	84%	79%

Table 8: *Percentage of correct answers for three CT concepts. Scores from Relkin et al.[20] were estimated from Figure 3. Scores from Lin et al. [14] were reported directly in Table 6; for comparison, we averaged the PRE scores of the plugged and unplugged group and converted them into percentages by dividing them by the maximum possible score for each concept (5 for Algorithms and 2 for Representation and Control Structures). Our scores were calculated as a mean PRE value of all groups (A1, A2, CG).*

Quantitative All the children correctly solved task 1 (choosing colors of the coding cards). Task 2, that required both choosing the color and the place, was correctly solved by 78% of children. The third task (conditional music reproduction) was solved by 56% of children, and the last task (the correct usage of the pink card) by 61%.

Overall, 36.1% of the children solved all tasks, 33.3% solved 3 of 4 tasks, 19.4% solved 2 tasks, and 11.1% solved only the first task.

Qualitative We present general observations along with the task specific ones.

General observations

SIMULATION. Overall, children did not rely on simulating Robotito's trajectory with their finger to determine the color and placement of the coding cards. However, in certain instances, simulating the path helped them to detect errors or determine the appropriate placement for the next color card.

CARD POSITION IN THE GRID. During tasks 2 and 4, some children correctly selected the colors of the cards but struggled with positioning them on the grid. For instance, they placed the second card too closely to the first one, causing the robot to change direction too soon. Alternatively, they positioned the card next to the robot's trajectory rather than directly within it.

AMOUNT OF CODING CARDS. Some children wanted to create trajectories that required more than the four color cards initially provided. This occurred when they aimed to create longer routes or used "redundant cards," placing two cards of the same color next to each other, even though the second card was unnecessary as it did not alter the robot's movement direction.

MOVING ROBOT IN SPACE. Some children struggled to understand how to use the color cards to move the robot in space. Two children placed the color cards adjacent to each other without considering the colors' meanings, effectively creating "a path from color tiles" rather than using the colors to indicate directions. In some instances, children believed the robot could move diagonally, although this movement is not supported by the robot. Additionally, when asked to demonstrate the robot's movement with their finger, some children indicated that the robot would change direction in locations where no color cards were present.

Task 2

COVERING ROBOTITO. In some cases, task 2 caused confusion among children regarding whether it was correct to cover Robotito's image with a color card. To complete the task, children needed to select a color card to initiate the robot's movements, placing it where the robot was drawn. Some children noted that it was impossible to put the color card beneath the robot and asked questions like, "It [the color card] goes above, or what?" Covering the robot with the first color card caused that the children had to lift it to check the color arrows on the robot and complete the task.

Task 3

This task caused different confusions and led to unexpected solutions.

COVERING COLOR CARDS. Many children tended to place the orange cards on top of the color cards that coded the U-shaped route. In these situations, the evaluator had to indicate that the color cards should not be covered.

IMPRECISE INSTRUCTIONS. In this task the children were asked "Where should we place the orange color cards so that Robotito makes sounds near the sun and is silent near the moon?" (see Figure 3 to visualize the task.) We observed that some children were unsure about what "near the sun/moon" meant in the context of the task. The idea behind the task was to turn on the music mode before the red card (the card closest to the sun) and turn it off before the blue card (the card closest to the moon). However, some children placed the orange cards after these cards, interpreting "near to" more broadly than we intended.

ORANGE CARD FUNCTION. Not all the children understood how the orange card changes Robotito's behavior. Some thought it would change the robot's movement direction, while others believed that the robot would only reproduce sounds when it passes over the card, using it to "make sounds" rather than to activate and deactivate the musical mode. In these cases, children placed only one card close to the sun to "make sounds near the sun." One child thought that to deactivate the musical mode, the orange card should be removed after the robot passed over it and made sounds.

INCORRECT PLACE. Some children tended to place both orange cards together between the color cards. They placed them in the first part of the U-shaped route (between the yellow and red card), in the bottom part (between the red and blue card), or even in the top part of the grid (between the yellow and the purple card), which was not part of the robot's route.

One child placed the orange card outside the grid and used it to cover the drawing of the sun.

Task 4

ALTERNATIVE CARDS. Some children ignored the fact that the robot should start on the pink card and instead tried to build alternative paths. These paths were not executed by the robot as they started with color cards placed next to the robot, but not in its way.

MOVEMENT SIMULATION. Many children simulated the "pink step" (moving one step in the yellow direction then one step in the green direction) to decide where to put the next color card. In some cases, they did not place the next card in the square where the "pink step" ended but instead placed it in the following square.

Summary of Robotito Test results Although all the children appeared to understand the rules that govern the robot's behaviors (task 1 was completed by all of the children), we identified several issues related to correctly positioning the color cards on the grid and misunderstandings robot's capabilities (e.g., moving diagonally, or changing direction without color cards).

We also observed issues arising from the on-paper nature of the test. Children were confused when they had to cover the robot image with the first card in task 2 or they tried to put orange cards on top of color cards in task 3—behaviors that were not observed during ER sessions focused on conditional music reproduction.

Some children used an extra set of color cards to build longer paths or placed "redundant" cards next to each other.

Test results indicated that task 3 was the most challenging for the children. Difficulties arose due to unclear instructions, a tendency to cover the color cards or place both orange cards together, and a misunderstanding of the orange card's function.

In the final task, we observed attempts to "ignore" the pink card, which was defined as the starting point of Robotito's route. In this task, children frequently used finger pointing to define where the robot would stop after executing the "pink step" and placed the following card there.

4.2.4 Correlations between tests

We analyzed the correlation between two validated CT tests to determine if overall scores and scores related to similar CT concepts showed a correlation. Additionally, we sought to validate whether the score of the Robotito Test correlates with the total score of any of the tests.

We calculated the Pearson correlation coefficient (PCC) for data following a normal distribution, and the Spearman rank correlation coefficient (rho) otherwise.

TechCheck-K and BCTt We analyzed the correlation between the general scores, as well as the PRE and POST scores, of both the BCTt and TechCheck-K tests.

The general scores of both tests showed a moderate (almost weak) correlation (rho=0.398). For PRE scores, the correlation was weak (rho=0.294), and the comparison of POST scores indicated a moderate (PCC=0.466) correlation.

Since both tests assess similar CT concepts—TechCheck-K assesses Algorithms while BCTt evaluates Sequences, and TechCheck-K evaluates Control Structures while BCTt evaluates If-then Conditionals—we examined whether the scores for these comparable concepts were correlated. In both cases, the correlation was weak, with rho=0.268 for Sequences and Algorithms and rho=0.189 for If-then Conditional and Control Structures (see Figure 5).

	SEQUENCES	IF	SIMPLE	NESTED	ALGORITHMS	REPRESENTATION	CONTROL
SEQUENCES	1.00	0.30	0.21	0.43	0.27	0.14	0.29
IF	0.30	1.00	0.26	0.35	0.12	0.14	0.19
SIMPLE	0.21	0.26	1.00	0.09	0.16	0.05	0.14
NESTED	0.43	0.35	0.09	1.00	0.07	0.16	0.22
ALGORITHMS	0.27	0.12	0.16	0.07	1.00	0.06	0.19
REPRESENTATION	0.14	0.14	0.05	0.16	0.06	1.00	0.02
CONTROL	0.29	0.19	0.14	0.22	0.19	0.02	1.00

n= 112

P

	SEQUENCES	IF	SIMPLE	NESTED	ALGORITHMS	REPRESENTATION	CONTROL
SEQUENCES		0.0015	0.0229	0.0000	0.0043	0.1497	0.0021
IF	0.0015		0.0048	0.0002	0.2126	0.1367	0.0456
SIMPLE	0.0229	0.0048		0.3433	0.1009	0.5768	0.1405
NESTED	0.0000	0.0002	0.3433		0.4440	0.0913	0.0183
ALGORITHMS	0.0043	0.2126	0.1009	0.4440		0.5613	0.0456
REPRESENTATION	0.1497	0.1367	0.5768	0.0913	0.5613		0.8727
CONTROL	0.0021	0.0456	0.1405	0.0183	0.0456	0.8727	

Figure 5: *Correlations between CT concepts. In the top part we see Spearman's rank correlation coefficients; in the bottom part the p-values.*

For the PRE scores, the correlations were weak (Sequences and Algorithms with rho=0.101 and If-then Conditional and Control Structures with rho=0.175) and statistically insignificant. The POST scores showed a moderate (almost weak) correlation (rho=0.381) in case of Sequences and Algorithms and a weak (rho=0.158), statistically insignificant correlation between Conditional and Control Structures.

BCTt and Robotito Test We conducted a correlation analysis between BCTt scores and Robotito Test scores. Since the Robotito Test was administered after ER activities, we compared its scores only with POST BCTt scores. We observed a moderate (rho=0.634) correlation.

We also analyzed the correlation of BCTt concepts scores with the Robotito Test. Sequences showed significant moderate (rho=0.61) correlation with the Robotito Test results.

We looked for patterns of similar behavior of Robotito Test scores and childrens' responses to questions related to Sequences. We observed that children with high Robotito Test scores (3-4 points) demonstrated improvement in their performance across the first 4 items related to Sequences (see Figure 6, specifically columns Seq1, Seq2, Seq3, and Seq4). These children (n=25) improved their answers in 31% of the cases, maintained correct answers in 57% of the items, maintained wrong answers in 8%, and worsened the answer in only 4%. Specifically, 48% (12 out of 25) of these children improved their scores on the first sequencing task, 24% on the second, 16%

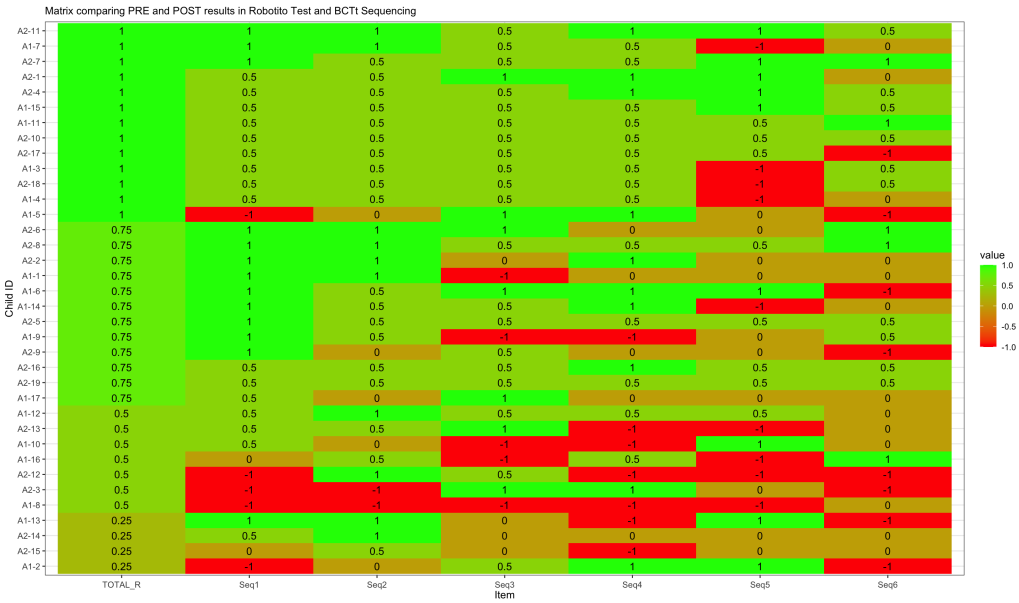


Figure 6: *Robotito Test scores and POST BCTt scores in 6 items related to Sequences. The first column presents Robotito's normalized score, while subsequent columns reflect POST scores coded as follows: 1 for correct answers if the child scored 0 in PRE BCTt test (indicating improvement from incorrect to correct in POST), 0.5 for correct answers if the child scored 1 in the PRE BCTt test (indicating maintenance of correct response), 0 for incorrect answers if the child scored 0 in the PRE BCTt test (indicating maintenance of incorrect response), or -1 for incorrect answers if the child scored 1 in the PRE BCTt test (indicating a decline in score due to incorrect response).*

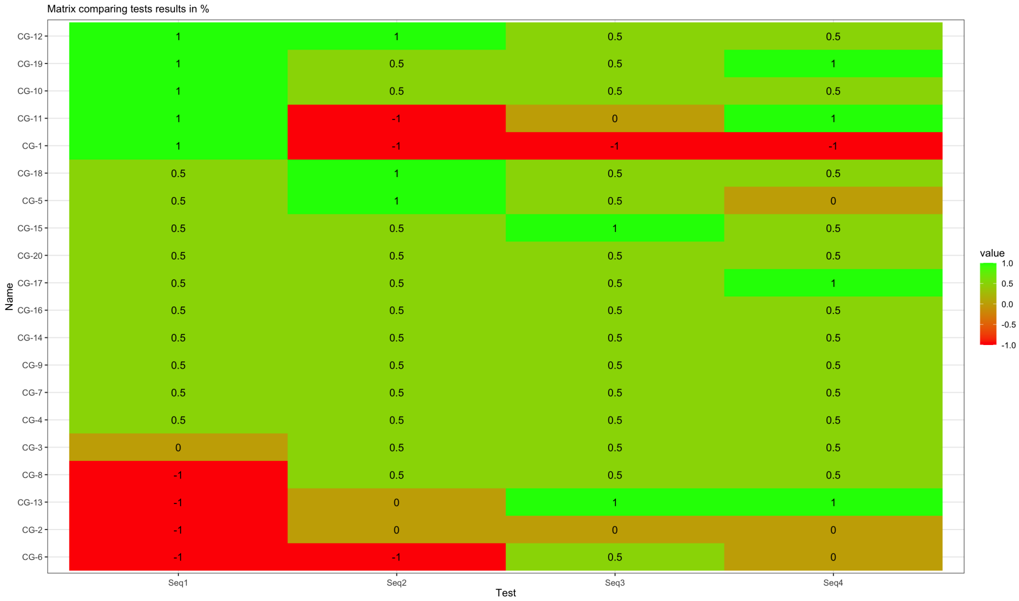


Figure 7: *Robotito Test scores and POST BCTt scores of CG in the first 4 items related to Sequences. The matrix scores were coded as in Figure 6.*

	Seq1	Seq2	Seq3	Seq4
AG-HR	100% (12/12)	67% (6/9)	67% (4/6)	67% (8/12)
CG	80% (4/5)	60% (3/5)	50% (2/4)	57% (4/7)

Table 9: *Improvement index for four initial BCTt Sequencing questions. Comparison between children with high Robotito Test scores from AG (AG-HR) and children from CG.*

on the third, and 32% on the fourth task. We hypothesize that the ER intervention influenced these improvements, as the CG showed lower improvement rates: 25% (5 out of 20) improvement on task 1, 15% on task 2, 10% on task 3, and 20% on the fourth task (see Figure 7).

As the possibility of improvement was lower in the CG due to its better performance in BCTt, we calculated an improvement index for each question. This was done by dividing the number of improved responses (correct responses in the POST-test following incorrect responses in the PRE-test) by the total number of possible improvements (scores of 0 in the PRE-test). This metric revealed that children with high Robotito Test scores outperformed the CG (see Table 9).

Also, the correlation between Robotito Test scores and the sum of the scores of the first four Sequences items showed strong (rho=0.701) correlation.

TechCheck-K and Robotito Test We identified a moderate (rho=0.457) correlation between TechCheck-K POST and Robotito Test scores.

In the case of Algorithms and Representation scores the correlations with the Robotito Test were weak (rho=0.092 and rho=-0.099) and not statistically significant. In contrast, Control Structures showed a significant moderate (rho=0.501) correlation.

Summary of correlations analysis Correlation analysis between TechCheck-K and BCTt scores showed no strong correlation. Similar CT concepts exhibited weak or moderate (almost weak) correlations.

The Robotito Test demonstrated a moderate correlation with both validated CT tests. Upon analyzing CT concepts, moderate correlations were observed between the Robotito Test and Sequenes (BCTt) and Control Structures (TechCheck-K) scores. However, the correlation between Robotito Test scores and the sum of the scores of the first four Sequences items showed strong (rho=0.701) correlation.

4.3 Discussion

The results of our analysis were surprising and left many open questions and future works.

4.3.1 Outperformance of the CG in BCTt

The comparison between AG and CG BCTt scores revealed a significant, but also surprising, difference in favor of the CG, which outperformed the AG, particularly the A1 class. Factorial ANOVA results confirmed a significant difference between AG and CG, especially between A1 and CG. However, there was no significant PRE-POST difference nor any interaction effect between time and group, indicating that the group variable is the sole factor explaining the score differences.

One potential explanation for the surprising difference in BCTt scores could be the class teachers' attitudes observed during interactions with the students. The CG teacher, who had more experience and an established trajectory in kindergarten, maintained the children's patience and focus during activities. She didn't raise her voice to manage inappropriate classroom behaviors. In contrast, the A1 teacher was younger and frequently used an elevated voice and threats to manage the classroom. We speculate that the CG teacher's approach could have influenced the children's attitude during the evaluation, helping them focus on the tasks and reflect calmly on their answers.

Given that the analysis of TechCheck-K scores did not reveal any differences between groups or classes, it is challenging to draw any definitive conclusions regarding differences in CT levels between the AG and CG.

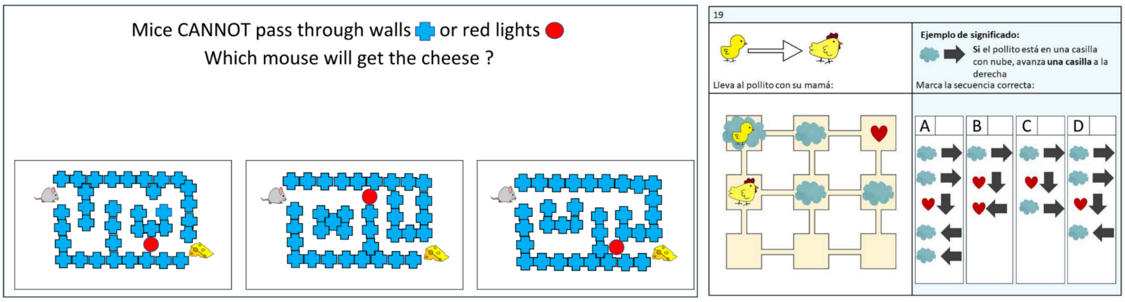


Figure 8: *Examples of items evaluating control structures in TechCheck-K (left) and BCTt (right).*

4.3.2 PRE-POST improvements within groups

Although we observed a generally better performance of the CG in the BCTt test, a PRE-POST comparison did not show any significant improvement in this group. There was no improvement in either BCTt or TechCheck-K scores, leading us to conclude that while the CG had a higher initial level of CT according to BCTt scores, it did not improve its performance in any test. In contrast, the AG, particularly the A2 class, significantly improved its BCTt scores. However, this improvement was not reflected in the TechCheck-K scores, making it difficult to determine if the AG truly improved its CT level.

Furthermore, at the level of specific CT concepts, we found disjunctive results. While the If-then Conditional score of BCTt improved in the CG, its corresponding concept in TechCheck-K, the Control Structures, showed an improvement in the other group, the AG.

The way the concept of control structures was represented in both tests (see Figure 8) could have influenced the results. Although both tests focused on if-then rules, they were incorporated differently. In the BCTt, children had to follow a programming sequence from top to bottom and analyze each if-then rule to determine if they could move the baby chicken in the grid. In contrast, in the TechCheck-K, the if-then rules were fixed and defined from the beginning of the task and had to be used to analyze the mazes. In the case of BCTt the rules could change as the chicken progresses through the grid. For example, in the answer A in Figure 8, standing on the cloud in the first rule makes the chicken move right, while in the last rule, it makes the chicken move left. This makes the task more complex than simply applying the same rule consistently. We observed differences in the percentage of correct answers that confirm BCTt tasks were more difficult: the If-then Conditional tasks of BCTt scored 50% (Item 7) and 29% (Item 8), while the Control Structures of TechCheck-K scored 84% (Item 8) and 73% (Item 9).

Based on this, we could explain the divergent progress results by assuming that the AG mastered the if-then concept at a less difficult level, while the CG mastered more complex if-then rules. This discrepancy could also account for the weak and statistically insignificant correlation between the two concepts revealed by the correlation analysis.

4.3.3 Comparison with other studies

Comparison with other studies showed strong correlation of our results with the correct response rate of each item reported in [30] and [18].

Although the scores of each item reported by Zapata et al. [30] are higher than our scores, this difference could be explained by the fact that their data encompassed children aged 5 to 8. Despite their higher scores, there is a high correlation with our results, confirming similar performance among the groups.

Some differences were observed when analyzing correct responses rates grouped by concepts. Since we compared only two numbers, no statistical analysis was possible; therefore, we could only express our subjective judgments. Regarding BCTt concept scores, they were similar to those reported by Zapata et al. [30] and Zapata-Cáceres and Fanchamps [31], showing particularly high similarity with the latter, which reports data for 5-year-old children.

Relkin and Bers [18] scores of each item were of similar magnitudes as our results, but the percentages of correct answers for each CT concept reported by Relkin et al. [20] were slightly lower for Algorithms and Representation and considerably lower for Control Structures compared to our results. Although our group showed a similar mean age (6.02 vs. 5.86 in [20]), they reported that the minimum age in their kindergarten group was 4. The inclusion of 4-year-old children in their sample could have influenced their scores. Lin et al.'s scores [14] were slightly higher than ours and Relkin et al.'s results, despite working with younger children who had a mean age of 5.51.

Given that the differences in scores were generally small and the correlation was strong, we consider that the groups performed similarly.

4.3.4 Correlation between the validated PC tests

Currently, there are only two validated CT tests, with few studies reporting their use in our age group, and no studies that report the use of both tests while analyzing the outcomes. As previously mentioned, we observed many contradictory results (see Table 10), raising the question of whether both tests measure the same ability.

Although we reduced the number of issues in each test, excluding concepts that the ER activities did not address, we expected the total scores of both tests to correlate, meaning that children performing well on one test would also perform similarly on the other. As Na et al. [15] pointed out, both tests are effective in discriminating between children with the same level of CT (to be precise between children with relatively low CT level), therefore we expected similar test results. Moreover, we anticipated that similar concepts would yield similar scores. However, statistical analyses revealed no strong correlation between the overall test scores or between scores of similar CT concepts (Sequences vs. Algorithms, If-then Conditional vs. Control Structures).

Comparison	BCTt	TechCheck-K
AG-CG scores (two-way t-test and factorial ANOVA)	X	-
A1-CG scores (two-way t-test and factorial ANOVA)	X	-
AG PRE-POST (paired t-test)	X	-
A2 PRE-POST (paired t-test)	X	-
If-then Conditional CG PRE-POST (Wilcoxon signed-rank test for paired data)	X	-
Control Structures AG PRE-POST (Wilcoxon signed-rank test for paired data)	-	X

Table 10: *Contradictory results across validated CT tests. Significant outcomes marked with the cross.*

The differences between BCTt and TechCheck-K results could be explained if our results showed a high discrepancy with other studies, indicating that our sample might not be representative. However, this is not the case, as our results showed a strong correlation with the studies already published by the authors of the tests.

These observations raise questions about the differences between the tests, which result in different scores within the same population. Further research is needed to delineate these differences, enabling researchers to select the most appropriate test for their specific research questions.

4.3.5 Correlations between the validated PC tests and Robotito Test

The Robotito Test results exhibit only weak to moderate correlations with the general scores and concept scores of validated computational CT tests[7]. The strongest correlations, though still moderate, were observed between the BCTt and the Robotito Test scores and between the Robotito Test scores and Sequences concept scores.

We noted that children with high Robotito Test scores demonstrated improvements in their responses to initial Sequences tasks. Statistical analysis revealed a strong correlation between the total score of the first four Sequences items and the Robotito Test scores. This finding suggests that the Robotito Test score may be indicative of children's abilities related to easy and moderate sequencing tasks.

4.3.6 ER activities impact

Validated CT tests showed contradictory results, making it challenging to draw definitive conclusions about the impact of ER activities with Robotito on children's CT development. According to BCTt results AG significantly improved its scores, but this finding was not confirmed by TechCheck-K results. Improvements related to Control Structures and If-then Conditional were also inconsistent.

[7]Since the Robotito Test was administered after the ER activities, we conducted the correlation analysis using the POST scores of the validated tests.

The Robotito Test results indicated that all the children understood the color-direction relationship essential for comprehending the robot's behavior, with more than half of the AG correctly solving each task. Additionally, children with high Robotito Test scores showed improvement in initial sequencing tasks of the BCTt. Students were able to plan trajectories for the robot, divide the route into smaller parts, and translate it into sequences of color cards. These observations suggest that the intervention was successful; however, the contradictory results from validated instruments prevent us from drawing definitive conclusions.

We identified two important aspects to consider for future activities to avoid surprises and confusion: study design and evaluation format. The unexpected outperformance of the CG and differences in performance across AG classes highlight the importance of random assignment to AG and CG for a fair comparison. Although random assignment was not feasible in this study due to the different class schedules, future studies should strive to implement it to complement the existing data. While this may be challenging in a kindergarten context, studies in informal settings could complement current results with additional data from more controlled experiments.

We also believe that the on-paper evaluation format used to validate knowledge about the concepts introduced in ER sessions may have caused some unnecessary confusions. During the Robotito Test, some children covered the color cards to activate or deactivate the musical mode, a behavior never observed with the real robot. Additionally, covering Robotito with the color card in task 2 was both surprising and problematic for the children. They were not accustomed to placing the color cards over the robot, and after doing so, they could not see the color-direction indicators on top of the robot. Consequently, they had to lift the first coding card to proceed with the task, which caused further confusion. Since the evaluation was conducted individually and by a single evaluator, using the real robot should not introduce significant differences. It may, however, result in a slightly longer evaluation time for each child.

4.3.7 Conditionals

Conditionals are present in both validated CT tests, indicating their significance in CT. Therefore, we believe they should be specifically targeted during ER activities aimed at developing CT.

However, there are scarce studies that report on ER activities that introduce conditionals to preschoolers and measure the impact. In our previous work [2], we identified only one empirical study [26] providing evidence of children mastering conditionals. Through additional literature review, we were able to identify another study reporting pre and post BCTt scores related to If-then Conditional [31]. Both studies confirm that preschoolers can solve tasks using conditional statements, although Zapata-Cáceres and Fanchamps [31] noted that this concept was not accessible to children younger than 5.

Given that there are only two empirical studies, with one admitting that teachers may have provided more help and scaffolding than intended during the evaluation [26], we believe our results contribute to the discussion on whether preschool children can understand and successfully use conditionals.

The results of the present study show promising findings related to the understanding and correct use of conditionals. Both validated tests indicated already in the PRE test, that children could solve tasks based on if-then statements. For BCTt items evaluating If-then Conditional, 50% of the children correctly solved Item 7, and 29% solved Item 8. TechCheck-K showed much higher scores for Control Structure items, with 84% correct answers for Item 8 and 73% for Item 9. Additionally, task 3 in the Robotito Test, which evaluates conditional music reproduction, was correctly solved by 56% of children. We believe this score could be higher, as issues such as imprecise instructions and problems related to the on-paper format of the evaluation caused some confusions.

Previous studies and our results appear to confirm that conditionals are accessible for children aged 5 to 6. We believe that ER interventions should include activities focused on conditionals to provide children with an opportunity to master more advanced programming concepts. This foundation will enable them to build complex algorithms in the future.

5 Conclusions

In this study, we reported two stages of developing conditionals in Robotito and implementing them in Robotito's curriculum.

In the first step, we explored various prototypes for incorporating conditionals into Robotito's programming. The prototypes included the color frame, musical mode, and split card, each offering distinct implementations of conditionals. Through multiple evaluations involving experienced teachers, we identified the strengths and weaknesses of each prototype. The color frame prototype was deemed too complex, while the musical mode and split card were found to be more accessible for preschoolers. The musical mode, in particular, was favored for introducing a new output modality—sound.

One of the tools used in our evaluations was Robotito's simulator. Teachers highlighted the simulator's value as a tool for introducing Robotito and working on programming tasks in a less distracting environment before using the real robot.

Taking into account the insights from the evaluation activities, we incorporated conditional music reproduction and Robotito's simulator based activities into Robotito curriculum and conducted a field study with level 5 preschoolers.

The activities were tested in two kindergarten classes, demonstrating that the children were able to understand and apply Robotito's rules to solve programming tasks during the classroom activities and in the final evaluation with Robotito Test. Also, potential improvements of Robotito Test were discussed.

The Robotito Test's evaluation of conditionals, along with the pre-test results of CT tests, showed that the children were capable of understanding and applying conditional statements to solve the proposed tasks. This suggests that conditionals can be introduced through ER activities already at the age of 5.

The analysis of scores from two validated CT tests revealed a lack of correlation and no similarities in scores between similar CT concepts. In some cases, the results were not only diverse but also contradictory. This highlights the need for further research to better understand the differences between the tests and the contexts in which they are appropriate.

Our study provides a unique data source that can be used for further analysis and comparisons with other experimental data. As our results support the accessibility of conditionals for 5 and 6-year-old children, we hope they will encourage the inclusion of this concept in ER curricula.

References

[1] Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, Maria Pascale, Camila Hergatacorzian, and Gonzalo Tejera. Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. In *Proceedings of the 21st annual acm interaction design and children conference*, pages 422–429, 2022.

[2] Ewelina Bakala, Anaclara Gerosa, Juan Pablo Hourcade, Gonzalo Tejera, Kerry Peterman, and Guillermo Trinidad. A systematic review of technologies to teach control structures in preschool education. *Frontiers in psychology*, 13:911057, 2022.

[3] Ewelina Bakala, Ana C. Pires, Mariana da Luz, María Pascale, Gonzalo Tejera, and Juan Pablo Hourcade. Coding with colors: Children's errors committed while programming robotito for the first time. In José Abdelnour Nocera, Marta Kristín Lárusdóttir, Helen Petrie, Antonio Piccinno, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2023*, pages 439–443, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-42293-5.

[4] Maurice Stevenson Bartlett. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 160(901):268–282, 1937.

[5] Héctor Belmar. Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science*, 4:997222, 2022.

[6] Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25, 2012.

[7] Ronald Aylmer Fisher, Ronald Aylmer Fisher, Statistiker Genetiker, Ronald Aylmer Fisher, Statistician Genetician, Great Britain, Ronald Aylmer Fisher, and Statisticien Généticien. *The design of experiments*, volume 21. Springer, 1966.

[8] Anaclara Gerosa, Víctor Koleszar, Gonzalo Tejera, Leonel Gómez-Sena, and Alejandra Carboni. Educational robotics intervention to foster computational thinking in preschoolers: Effects of children's task engagement. *Frontiers in Psychology*, 13:904761, 2022.

[9] Katerina V Glezou. Fostering computational thinking and creativity in early childhood education: Play-learn-construct-program-collaborate. In *Mobile learning applications in early childhood education*, pages 324–347. IGI Global, 2020.

[10] Shuchi Grover and Roy Pea. Computational thinking in k-12: A review of the state of the field. *Educational researcher*, 42(1):38–43, 2013.

[11] Shuchi Grover and Roy Pea. Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, 19(1):19–38, 2018.

[12] Dennis E Hinkle, William Wiersma, Stephen G Jurs, et al. *Applied statistics for the behavioral sciences*, volume 663. Houghton Mifflin Boston, 2003.

[13] David C Howell. *Statistical methods for psychology*. PWS-Kent Publishing Co, 1992.

[14] Yan Lin, Hongjian Liao, Suxian Weng, and Wanqi Dong. Comparing the effects of plugged-in and unplugged activities on computational thinking development in young children. *Education and Information Technologies*, pages 1–34, 2023.

[15] Chungsoo Na, Jody Clarke-Midura, Jessica Shumway, Wilhelmina van Dijk, and Victor R Lee. Validating a performance assessment of computational thinking for early childhood using item response theory. *International Journal of Child-Computer Interaction*, 40:100650, 2024.

[16] Stamatios Papadakis and Michail Kalogiannakis. Learning computational thinking development in young children with bee-bot educational robotics. In *Handbook of research on tools for teaching computational thinking in P-12 education*, pages 289–309. IGI Global, 2020.

[17] Stamatios Papadakis, Michail Kalogiannakis, and Nicholas Zaranis. Developing fundamental programming concepts and computational thinking with scratchjr in preschool education: a case study. *International Journal of Mobile Learning and Organisation*, 10(3):187–202, 2016.

[18] Emily Relkin and Marina Bers. Techcheck-k: A measure of computational thinking for kindergarten children. In *2021 IEEE global engineering education conference (EDUCON)*, pages 1696–1702. IEEE, 2021.

[19] Emily Relkin, Laura de Ruiter, and Marina Umaschi Bers. TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, 29(4):482–498, 2020. ISSN 1573-1839. doi: 10.1007/s10956-020-09831-x. URL https://doi.org/10.1007/s10956-020-09831-x.

[20] Emily Relkin, Sara K Johnson, and Marina U Bers. A normative analysis of the techcheck computational thinking assessment. *Educational Technology & Society*, 26(2):118–130, 2023.

[21] Evgenia Roussou and Maria Rangoussi. On the use of robotics for the development of computational thinking in kindergarten: Educational intervention and evaluation. In Munir Merdan, Wilfried Lepuschitz, Gottfried Koppensteiner, Richard Balogh, and David Obdržálek, editors, *Robotics in Education*, pages 31–44, Cham, 2020. Springer International Publishing. ISBN 978-3-030-26945-6.

[22] Bocconi S, Chioccariello A, Kampylis P, Dagienė V, Wastiau P, Engelhardt K, Earp J, Horvath MA, Jasutė E, Malagoli C, Masiulionytė-Dagienė V, Stupurienė G, Inamorato Dos Santos A, Cachia R, Giannoutsou N, and Punie Y. Reviewing computational thinking in compulsory education. Scientific analysis or review KJ-06-22-069-EN-N (online), Luxembourg (Luxembourg), 2022.

[23] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

[24] Valerie J Shute, Chen Sun, and Jodi Asbell-Clarke. Demystifying computational thinking. *Educational research review*, 22:142–158, 2017.

[25] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

[26] Amanda Sullivan and Marina Umaschi Bers. Dancing robots: integrating art, music, and robotics in singapore's early childhood centers. *International Journal of Technology and Design Education*, 28:325–346, 2018.

[27] Gonzalo Tejera, Guillermo Amorin, Andrés Sere, Nicolás Capricho, Pablo Margenat, and Jorge Visca. Robotito: programming robots from preschool to undergraduate school level. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 296–301. IEEE, 2019.

[28] Thomas D Wickens and Geoffrey Keppel. *Design and analysis: A researcher's handbook*. Pearson Prentice-Hall Upper Saddle River, NJ, 2004.

[29] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer, 1992.

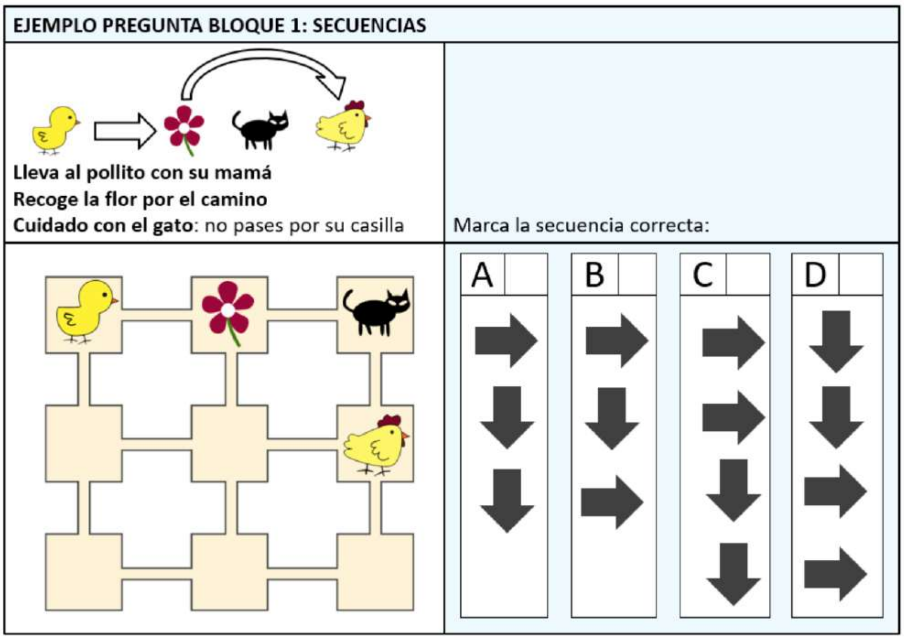
[30] María Zapata-Cáceres, Estefanía Martín-Barroso, and Marcos Román-González. Computational thinking test for beginners: Design and content validation. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1905–1914. IEEE, 2020.

[31] María Zapata-Cáceres, Nardie Fanchamps, et al. Using the beginners computational thinking test to measure development on computational concepts among preschoolers. In *Proceedings of the Fifth Asia Pacific Society for Computers in Education International Conference on Computational Thinking and STEM Education*, 2021.

Appendices

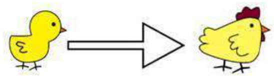
A BCTt tailored for our study

Madrid – febrero 2020 BCTt- Test de Pensamiento Computacional



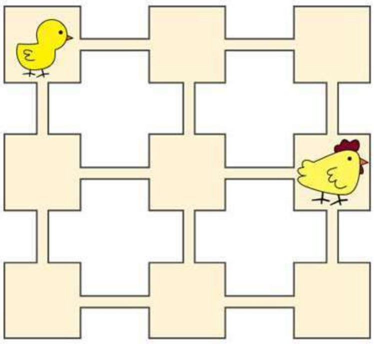
maria.zapata@urjc.es, estefania.martin@urjc.es 11

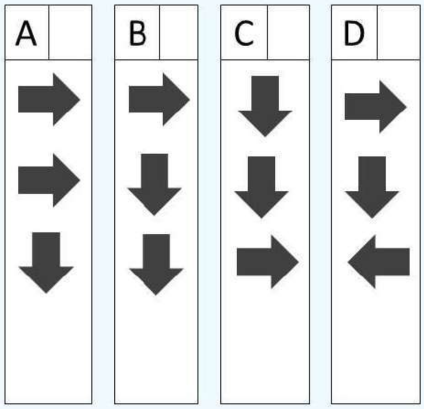
1



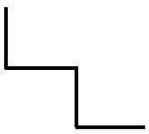
Lleva al pollito con su mamá:

Marca la secuencia correcta:



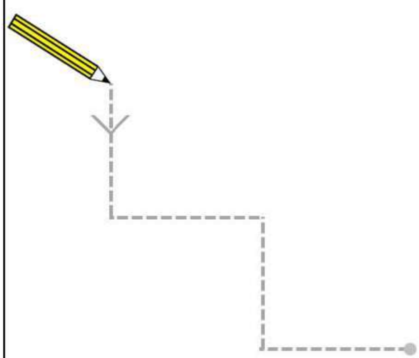


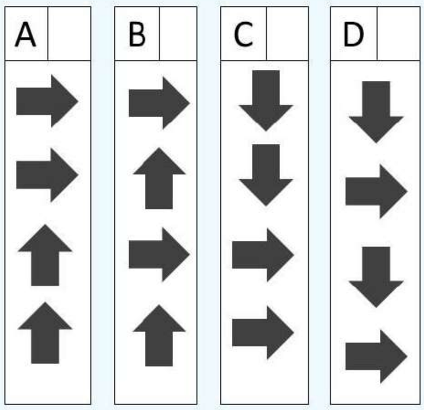
2



Dibuja la figura:

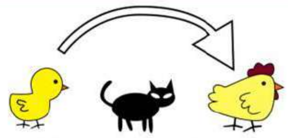
Marca la secuencia correcta:





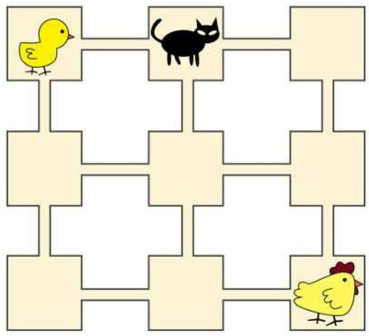
maria.zapata@urjc.es, estefania.martin@urjc.es

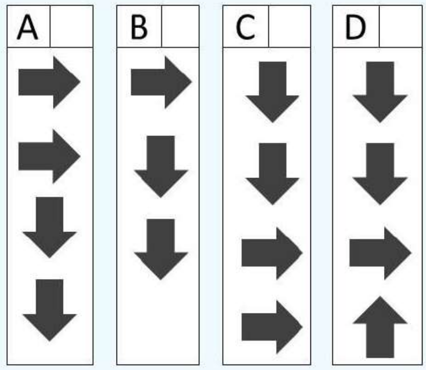
3



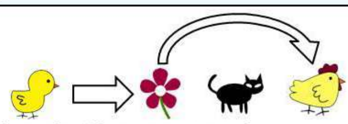
Lleva al pollito con su mamá.
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:



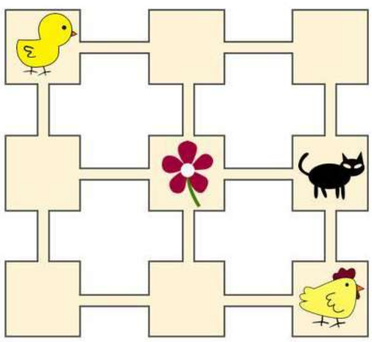


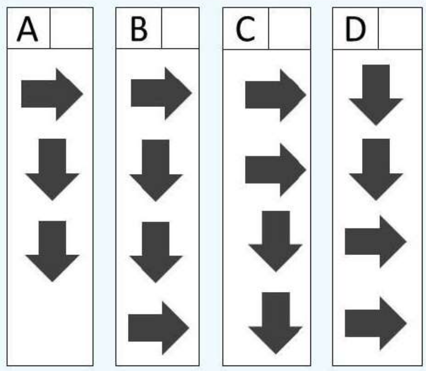
4



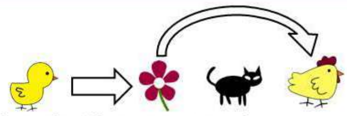
Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

Marca la secuencia correcta:



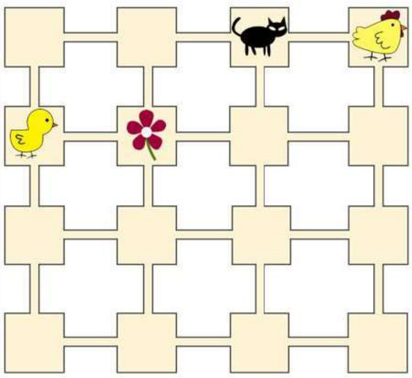


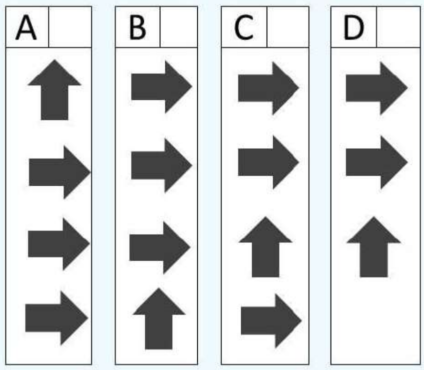
5



Lleva al pollito con su mamá
Recoge la flor por el camino
Cuidado con el gato: no pases por su casilla

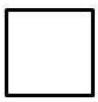
Marca la secuencia correcta:



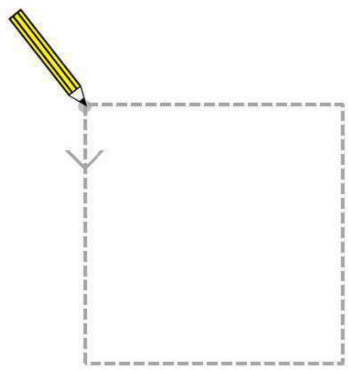


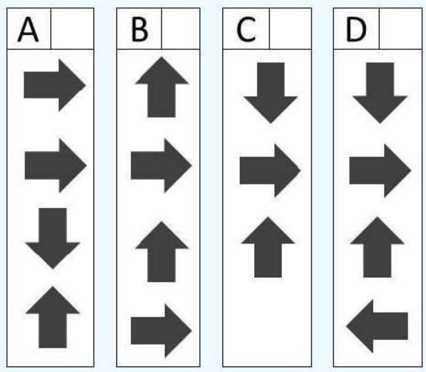
6

Dibuja la figura:



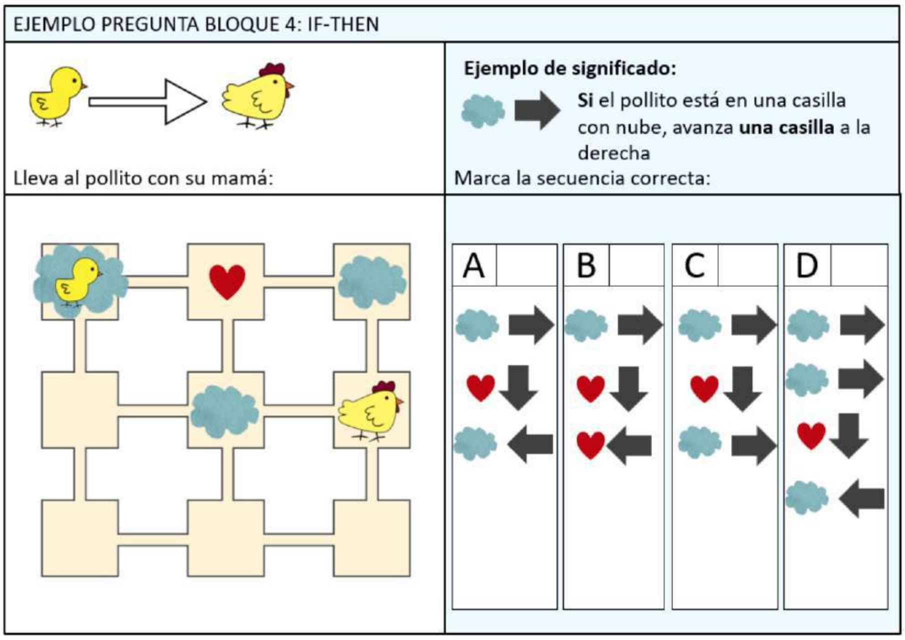
Marca la secuencia correcta:





maria.zapata@urjc.es, estefania.martin@urjc.es

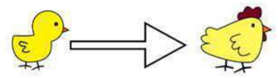
Madrid – febrero 2020 BCTt- Test de Pensamiento Computacional



11

maria.zapata@urjc.es, estefania.martin@urjc.es

7

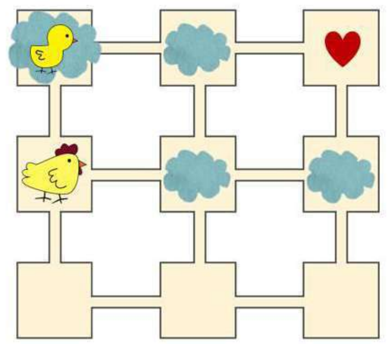


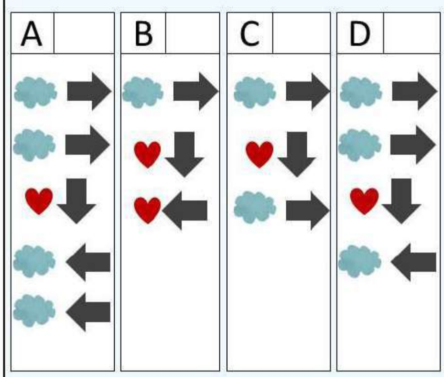
Lleva al pollito con su mamá:

Ejemplo de significado:

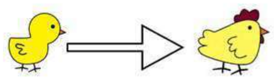
Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

Marca la secuencia correcta:





8

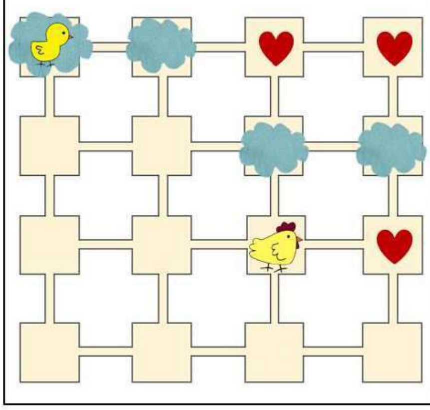


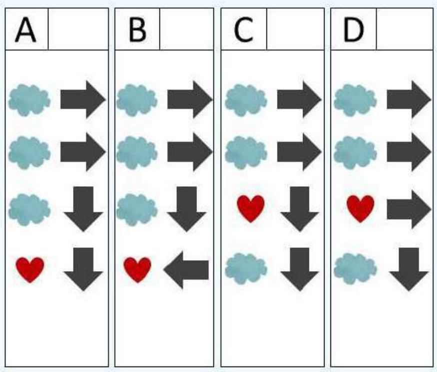
Lleva al pollito con su mamá:

Ejemplo de significado:

Si el pollito está en una casilla con nube, avanza **una casilla** a la derecha

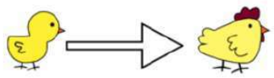
Marca la secuencia correcta:





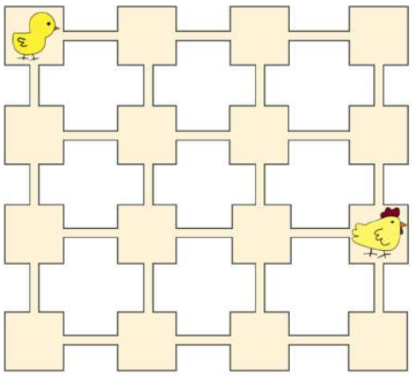
maria.zapata@urjc.es, estefania.martin@urjc.es

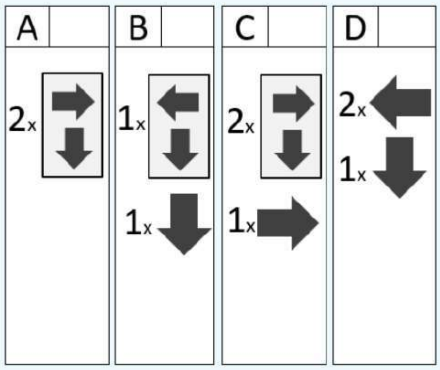
EJEMPLO PREGUNTA BLOQUE 3: BUCLE ANIDADO



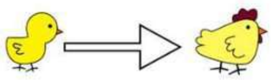
Lleva al pollito con su mamá:

Marca la secuencia correcta:





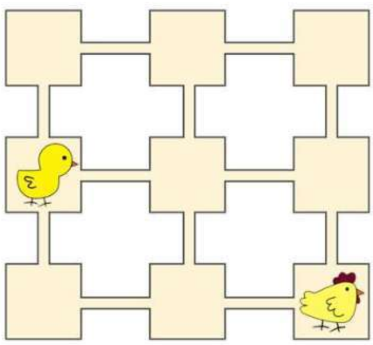
9

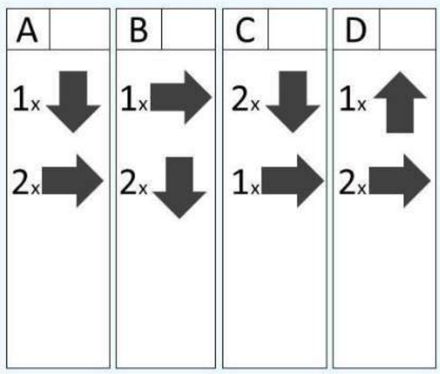


Lleva al pollito con su mamá:

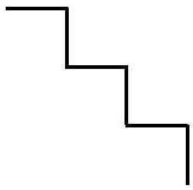
Marca la secuencia correcta:

Marca la secuencia correcta:



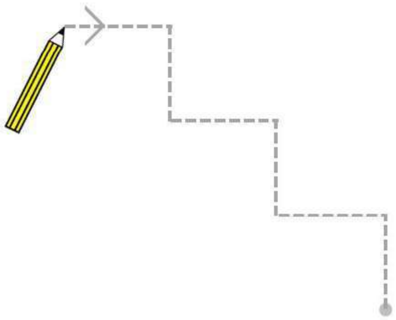


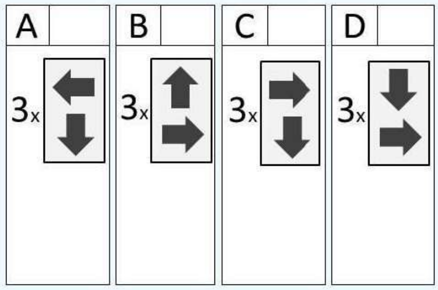
10



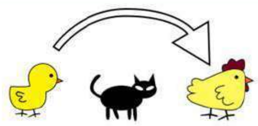
Dibuja la figura:

Marca la secuencia correcta:



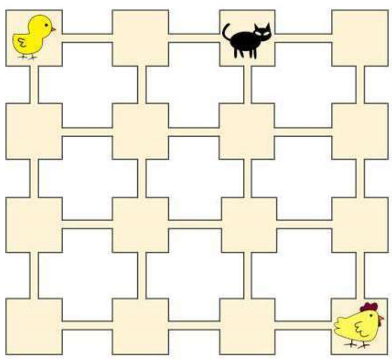


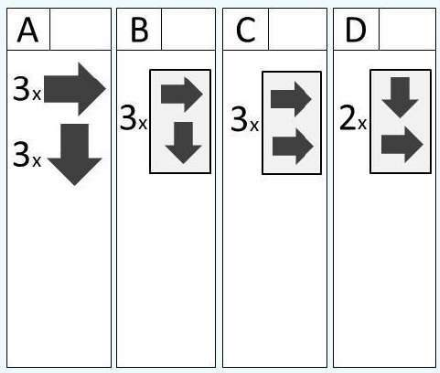
11



Lleva al pollito con su mamá.
Cuidado con el gato.

Marca la secuencia correcta:



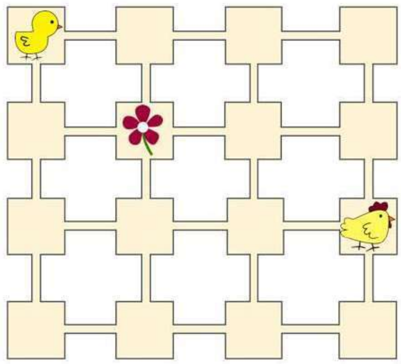


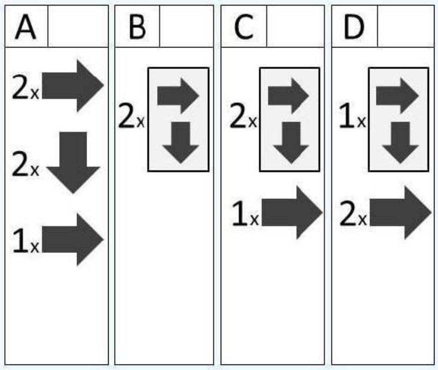
maria.zapata@urjc.es, estefania.martin@urjc.es

12

Lleva al pollito con su mamá
Recoge la flor por el camino

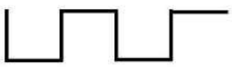
Marca la secuencia correcta:



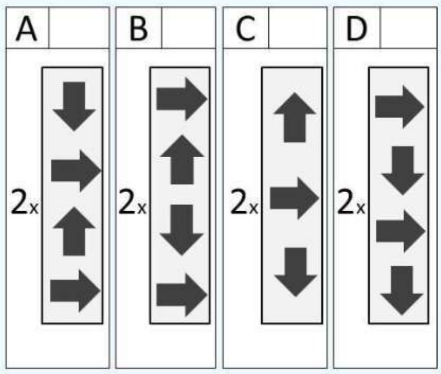


13

Dibuja la figura:



Marca la secuencia correcta:



maria.zapata@urjc.es, estefania.martin@urjc.es

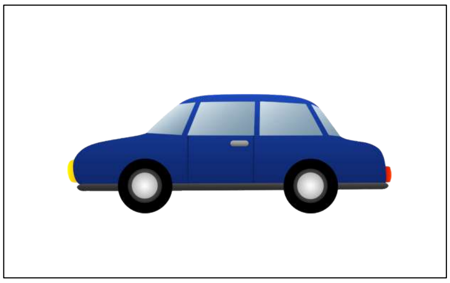
B TechCheck-K tailored for our study

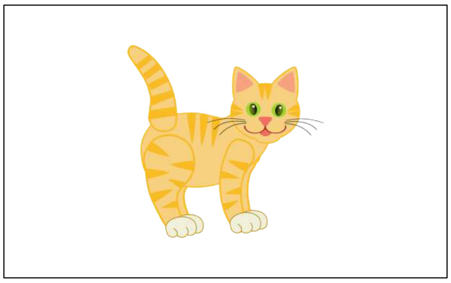
-Práctica 1-

¿Qué cosa se puede comer?

-Práctica 2-

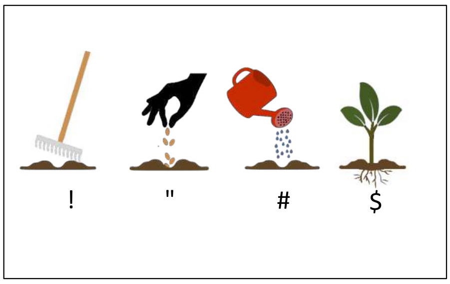
¿Cuál es un animal?

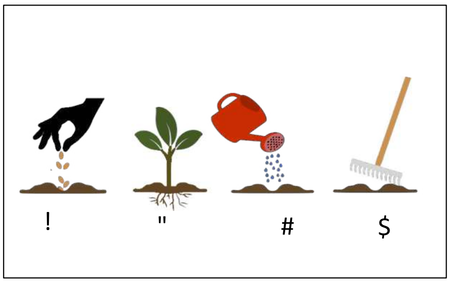


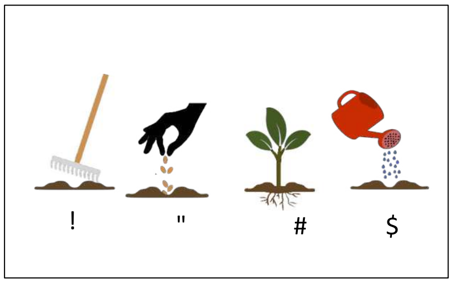


-1-

¿Cuál es el orden correcto para cultivar una planta?



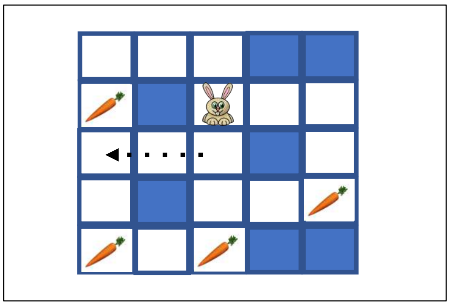


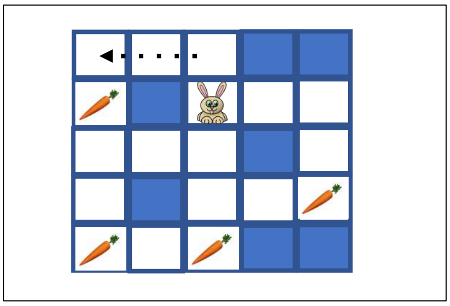


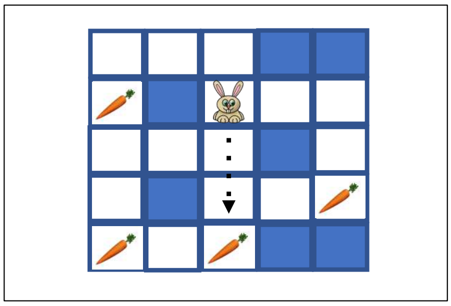
-2-

El conejito solo puede saltar un cuadrado blanco a la vez.

¿Cuál es la forma más rápida para que el conejito consiga UNA zanahoria?

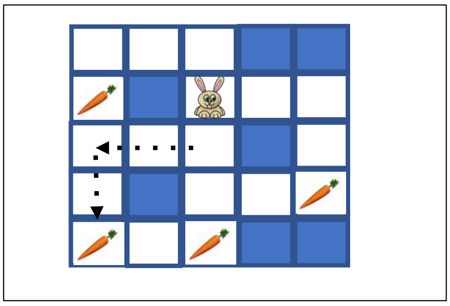


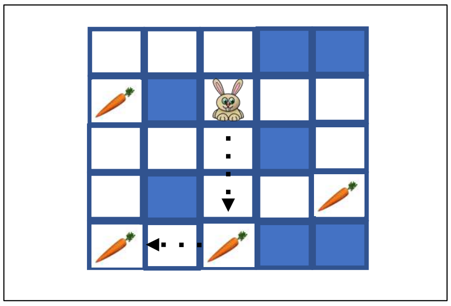


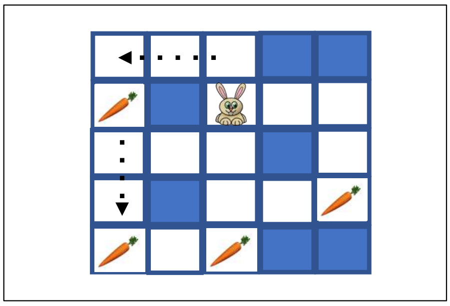


-3-

El conejito solo puede saltar un cuadrado blanco a la vez.
¿Cuál es la forma más rápida para que el conejito consiga DOS zanahorias?

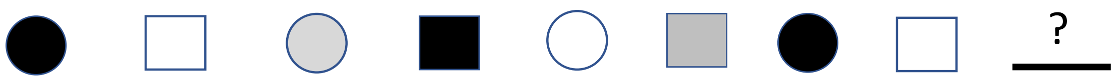


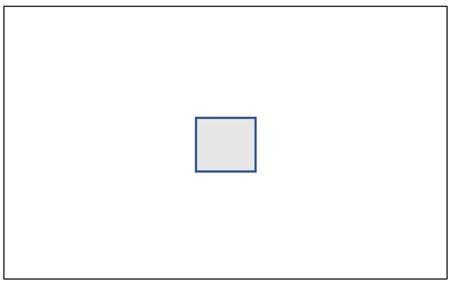


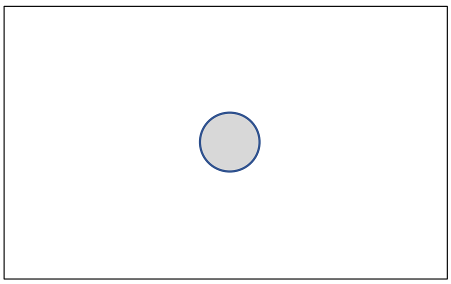


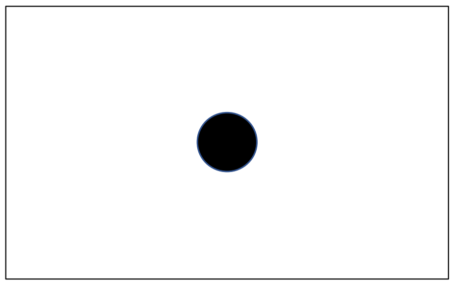
-4-

¿Qué viene a continuación?



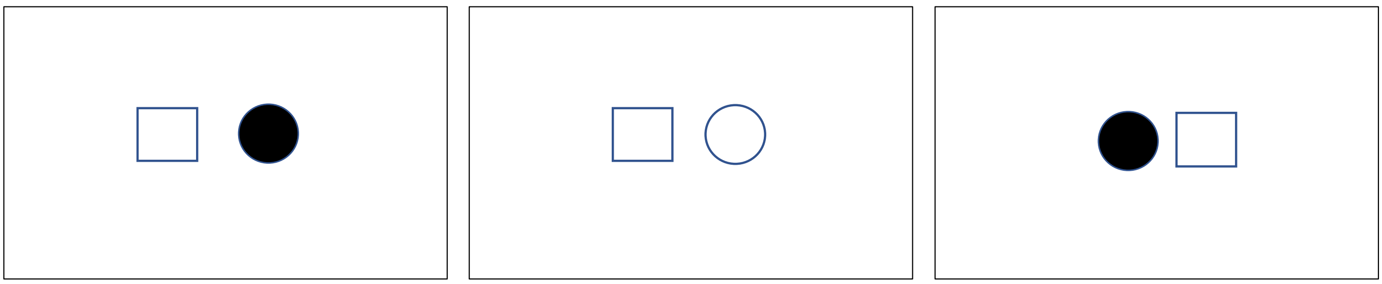




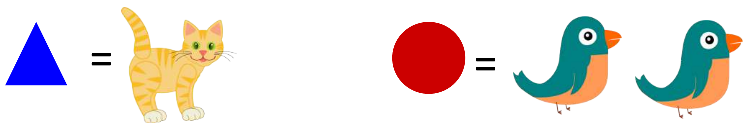


-5-

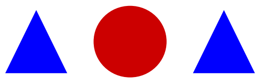
¿Qué viene a continuación?

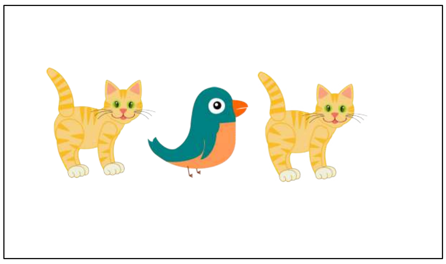


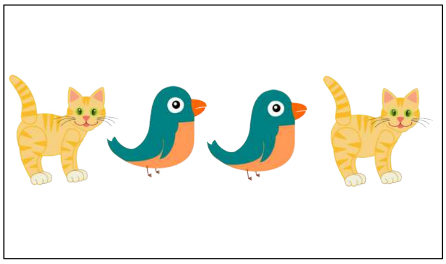
-6-

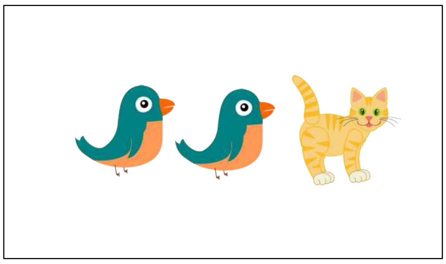


Si un triángulo representa un gato y un círculo representa dos pájaros, ¿qué representan estas tres formas?



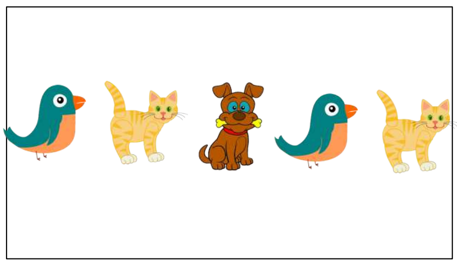


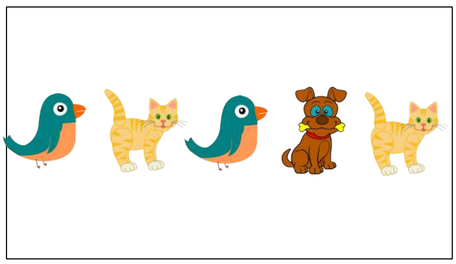


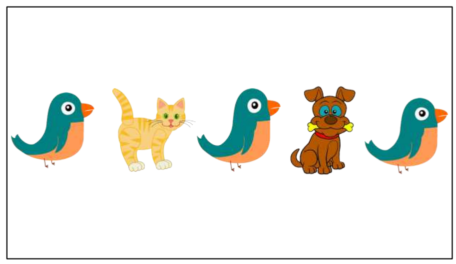


-7-

Un círculo representa un pájaro y un gato. Un cuadrado representa un perro y un pájaro. ¿Qué representan estas figuras?



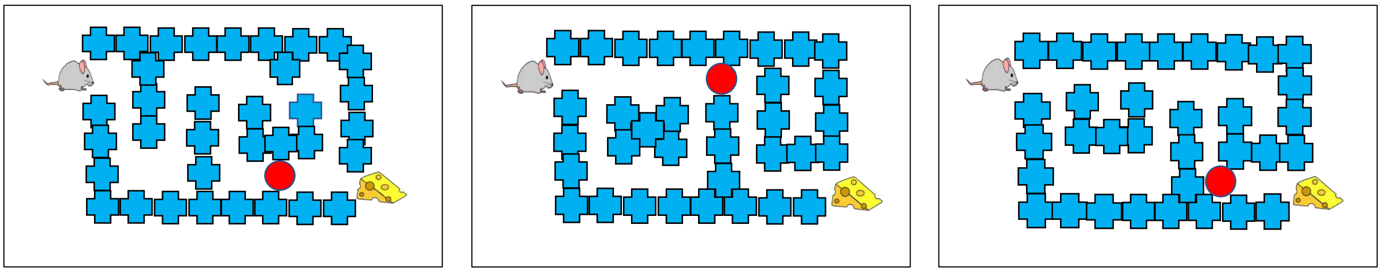




-8-

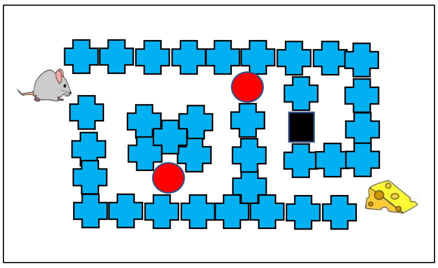
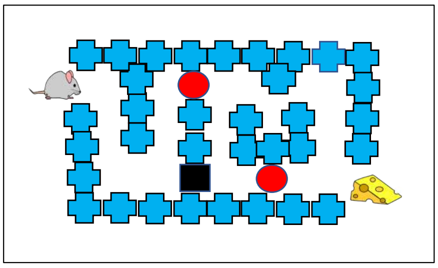
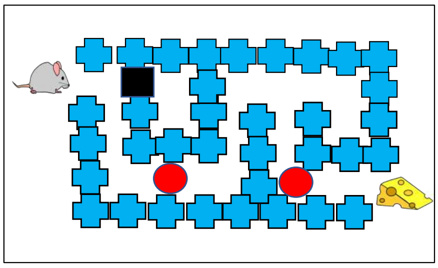
Los ratones NO pueden atravesar paredes o luces rojas

¿Qué ratón conseguirá el queso?



-9-

Los ratones NO pueden atravesar paredes o luces rojas, pero SI pueden atravesar túneles negros. ¿Qué ratón conseguirá el queso?

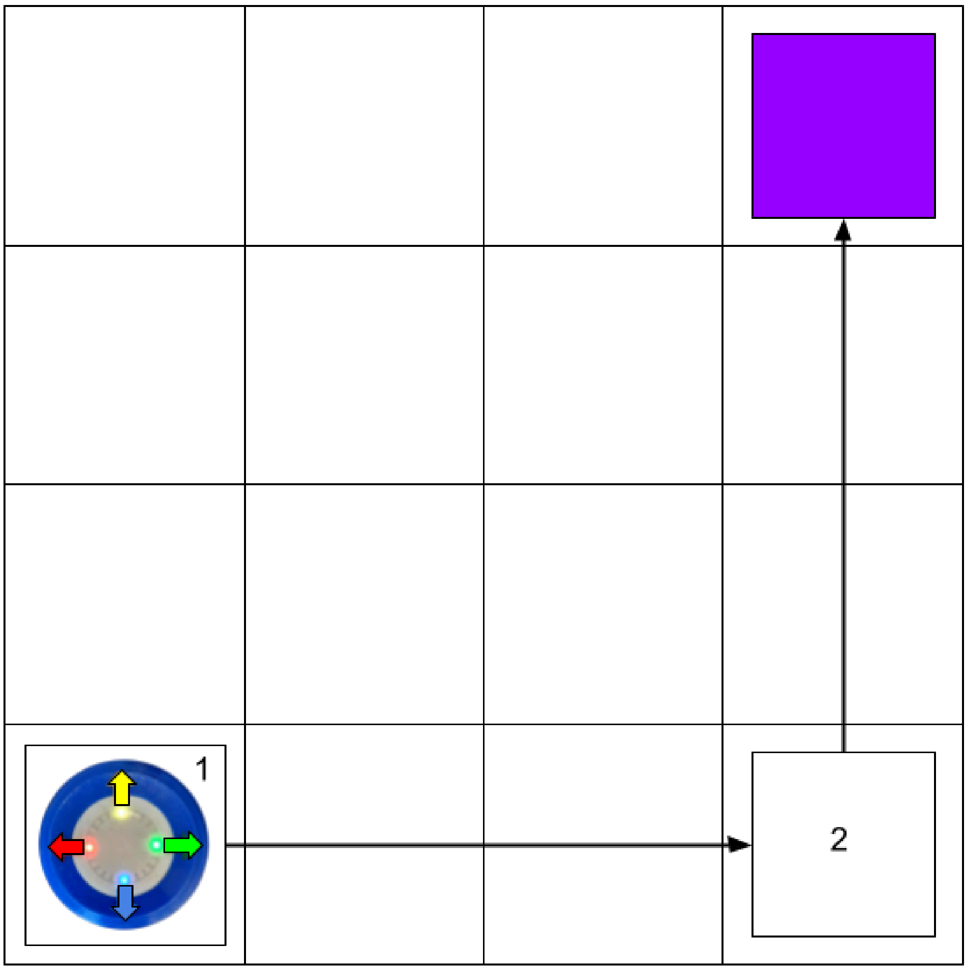
C Intervention's rubric

- Activity context (Options: formal, informal): FORMAL
- Number of participants (Number): 56
- Modality of work (Options: individual, group-based, mixed): MIXED Number of participants per group (Number): HALF CLASS OR 1 to 3
- Type of activity (Options: goal-oriented, open-ended, mixed): MAINLY GOAL-ORIENTED
- Free-play (Options: yes, no): NO
- Activity duration
  - Number of sessions (Number): 8
  - Session duration (Number in minutes): 30 to 75 minutes
  - Sessions' frequency (Number per week): 0 to 3
- Adults guiding the activities
  - Number (Number): 1 or 2
  - Adult–child ratio (Number): 1-WHOLE CLASS, 1-HALF CLASS, 1-SMALL GROUP (1 TO 3)
- Scaffolding provided
  - Adults' support (String with description of the type and degree of support): GUIDING QUESTIONS
  - Narrative (String with description): NO
  - Auxiliary objects (String with description): 4X4 MAT, COLOR CARDS, BLOCKING CARDS
  - Embodied examples (String with description): PLAYING TO BE ROBOTITO
  - Others (String with description): ROBOTITO SIMULATOR USED ON TV
- Unplugged activities (String with description): DRAWING ROBOTITO, ON-PAPER TASKS (COLORING OR DRAWING CODING CARDS)
- Explicit error detection (Options: yes, no): NO
- Relation with other domains (String with description): NO
- Communicating, sharing, and creating community (Options: yes, no): AT THE BEGINNING OF THE FIRST THREE SESSIONS

D Robotito Test

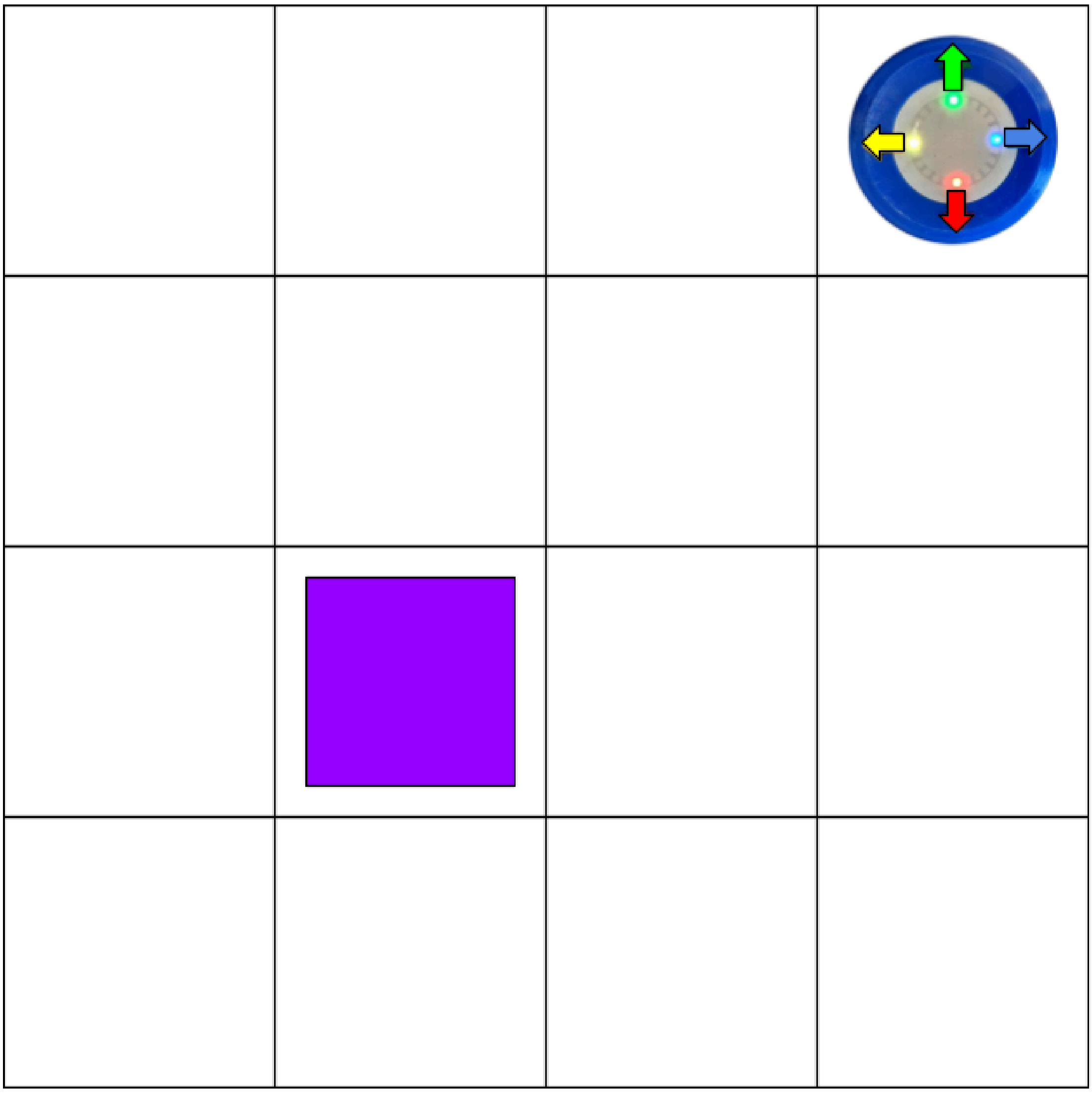
Tarea 1

¿Qué color (amarilla, roja, verde, azul) deben tener las tarjetas 1 y 2 para que el Robotito llegue a la tarjeta violeta?



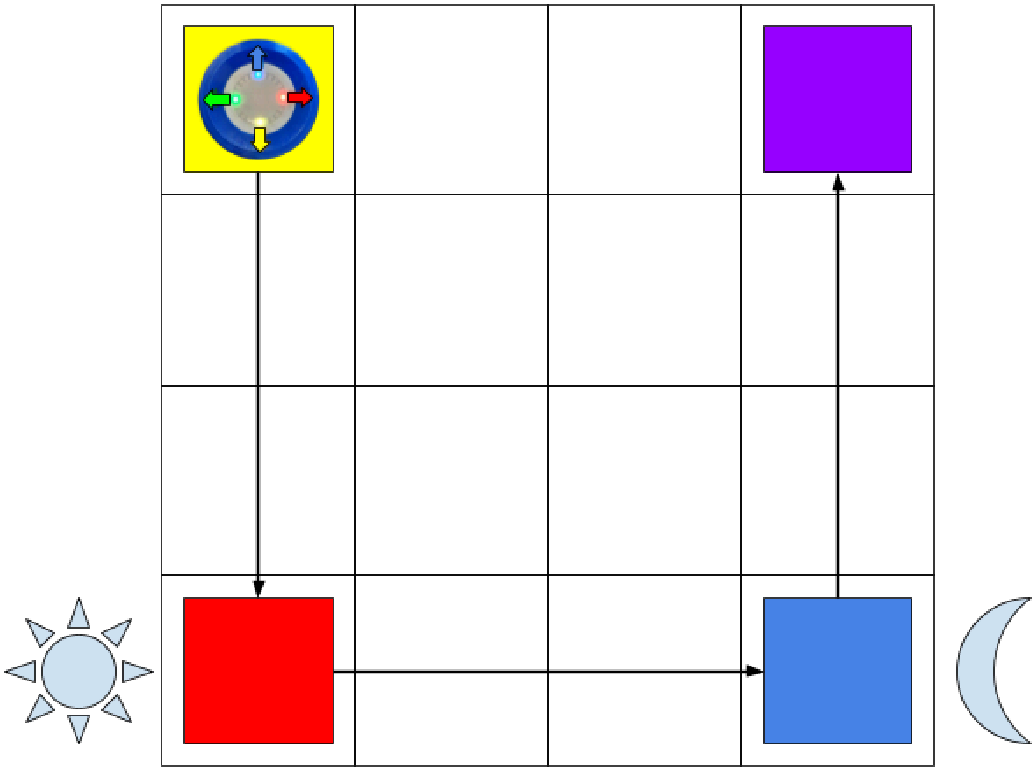
Tarea 2

Poné las tarjetas para que el Robotito llegue a la tarjeta violeta. Selecciona de qué color deben ser y en qué lugar ponerlas.



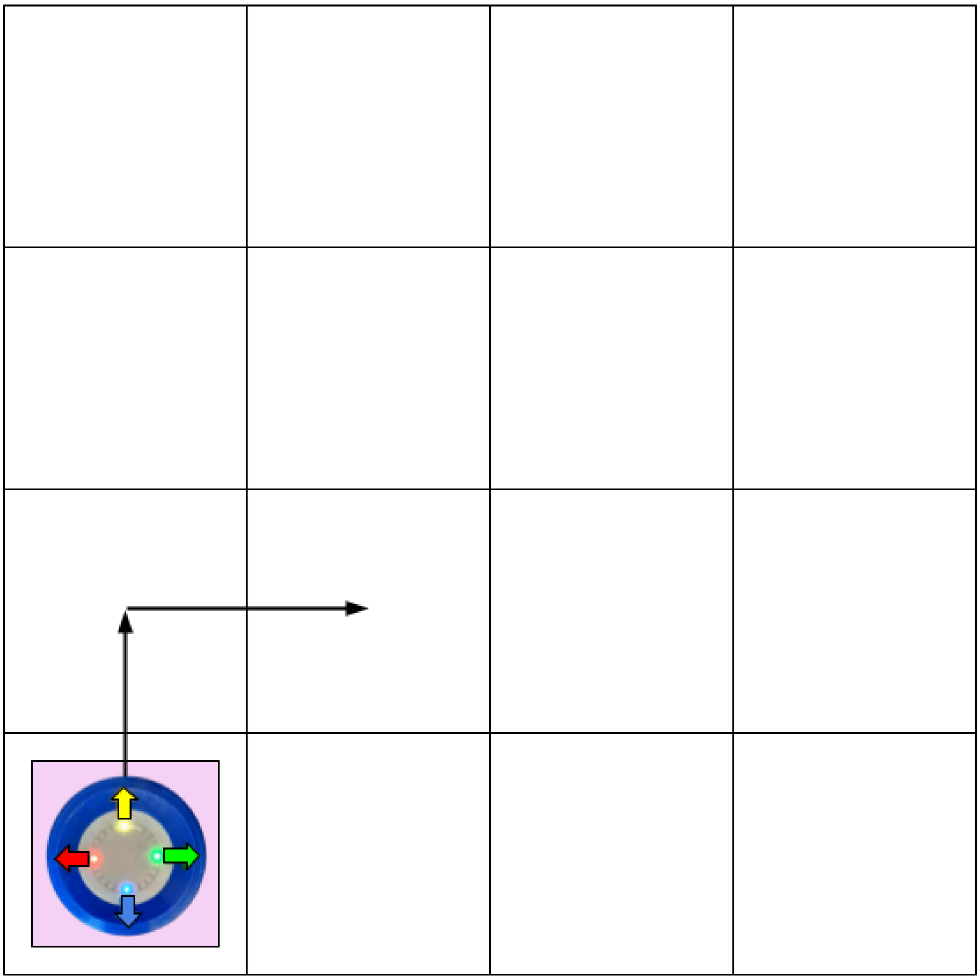
Tarea 3

El Robotito va a dar un paseo para llegar a la tarjeta violeta. En el camino debe pasar haciendo sonido cerca del sol y sin sonido cerca de la luna. ¿Dónde debemos colocar las tarjetas de color naranja para que el Robotito haga sonidos cerca del sol y no los haga cerca de la luna?

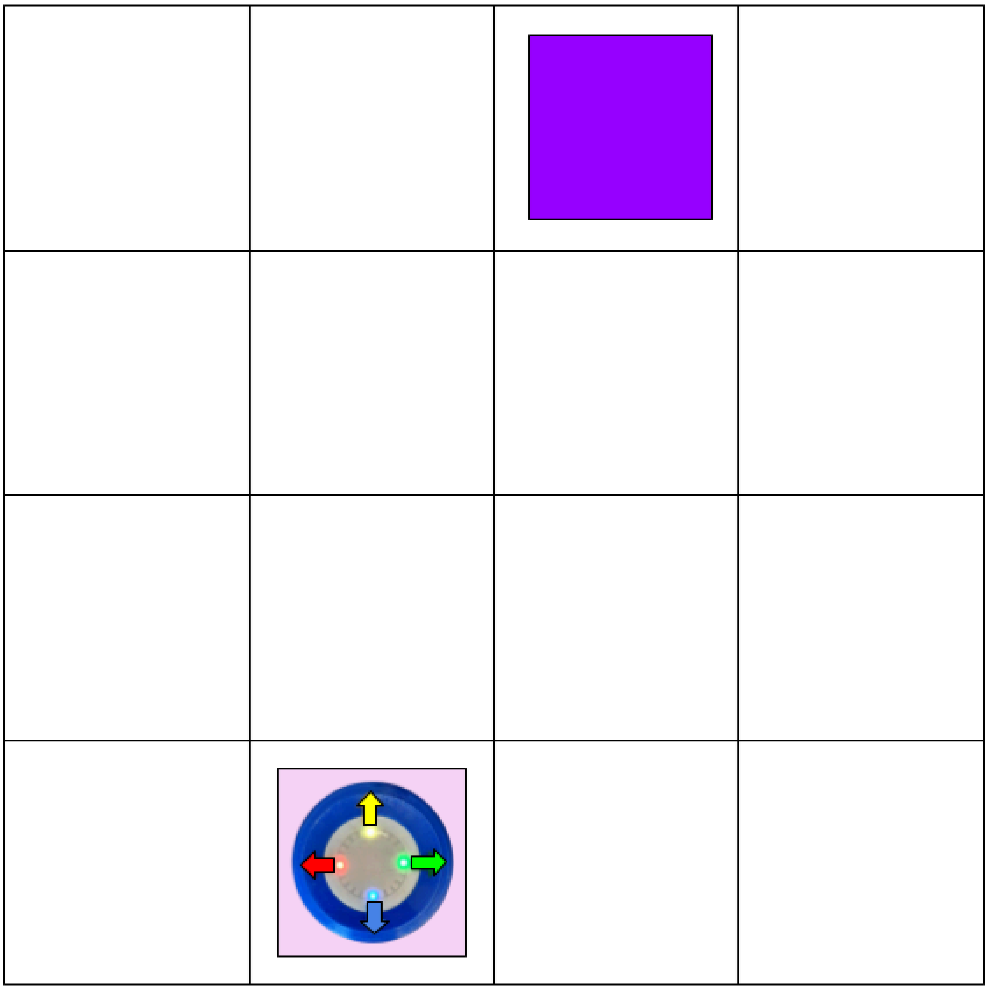


Tarea 4

El Robotito usa la tarjeta rosada para ir adelante y para el costado derecho.



Robotito empieza en la tarjeta rosada, ¿qué tarjeta hay que agregar y dónde para que llegue a la tarjeta violeta?



E Activities

Activity #1 Total time: 40 minutes. We asked the children to reflect on what the robots are and discussed with them the ideas. Each child explored Robotito (robot turned off) and we talked about what they observed (its parts, materials). We turned on Robotito and thought about how to control it. We explored how it moves with yellow, red, green and blue color cards and fixed the color paper arrows on the top of the robot to indicate the directions in which the robot moves after sensing a specific color.

Activity #2 Total time: 60 minutes. First, we reviewed the components of Robotito and its responses to color cards. Next, we explained that the activity would be conducted in two groups: one group would draw, while the other would role-play as Robotito, with roles switching afterward. For the drawing activity we provided three Robotitos: two normal robots and one without the shell to observe the inner parts of the robot.

The group that was playing to be Robotito was divided in pairs. One pair acted in front of the rest of the children that observed from their chairs. One child from each pair was acting as Robotito, the other as a programmer that places the color cards on the floor to move the robot. The idea was to direct the robot without hitting the furniture or the classmates. After a while of playing, the children switched the roles. After one pair went through playing robot and programmer, the next pair was called to perform in front of the others.

Figure 9: *Three moments of Activity #3. From left to right: child selecting were to place the robot with fixed orientation to reach the opposite site of the mat; child selecting a color card that will be placed on the floor to avoid that the robot escape from the circle formed by the children; child rotating the robot to reach the purple card.*

Activity #3 Total time: 60 minutes. We discussed the drawings done in the previous session. We splitted the group in two. Each group worked with one researcher on the same activity—the children were divided in two teams; each team sat on the opposite side of the mat. In the first part of the activity the children had to direct the robot to

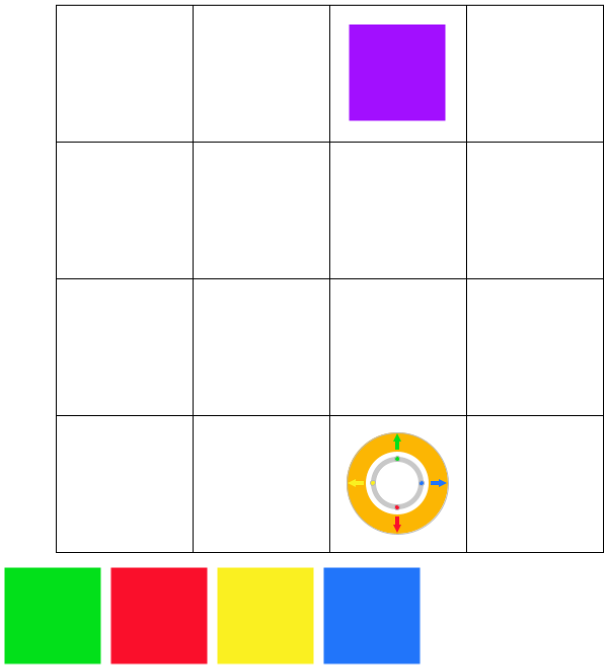


Figure 10: *Screenshot of an Android application simulating programming activity with Robotito.*

the opposite team by choosing the color card to put the robot on, as the orientation of the robot was defined by the researcher and could not be changed (see Figure 9).

Then we discovered a new card—a purple card that makes the robot turn all the lights purple and spin on the spot. This card was used as a destination card in the second part of the activity. The purple card was placed next to the opposite team and color cards were placed in front of the team that was handling Robotito. The child that was on task had to reach the purple card by rotating the robot and putting it in the correct place on the mat (see Figure 9). Each child did both—chose the color card and rotated the robot.

In the final part of the session we proposed a more open-ended activity in which the children were sitting in a circle and one child was putting color cards in the robot's path to prevent it from leaving the circle (see Figure 9). After the child selected the color of the card and put it in the robot's trajectory, the color cards were passed to the next child.

Activity #4 Total time: 40 minutes. In this activity the children interacted with an Android application that was simulating Robotito, 4 x 4 mat and color cards (see Figure 10). We first explained the app on TV to the whole group and solved together with children some example tasks. The children worked in pairs changing the person

Figure 11: *Children observing the robot's behavior after passing over the orange card.*

that is in charge of programming on the tablet. The programmer had to choose color cards to guide the robot to the purple card. Once reached the purple card, the child drew a smiling face on the A4 paper sheet to mark that the task was fulfilled and passed the tablet to its partner that proceeded with the next task.

Activity #5 Total time: 40 minutes. The class was divided into two groups. The first group worked in pairs on on-paper tasks in which the children had to paint already fixed coding card with the the right colors to make the robot reach the purple card, or define the place and the color of the cards that direct the robot to the purple card and draw them on the paper grid (see Appendix F).

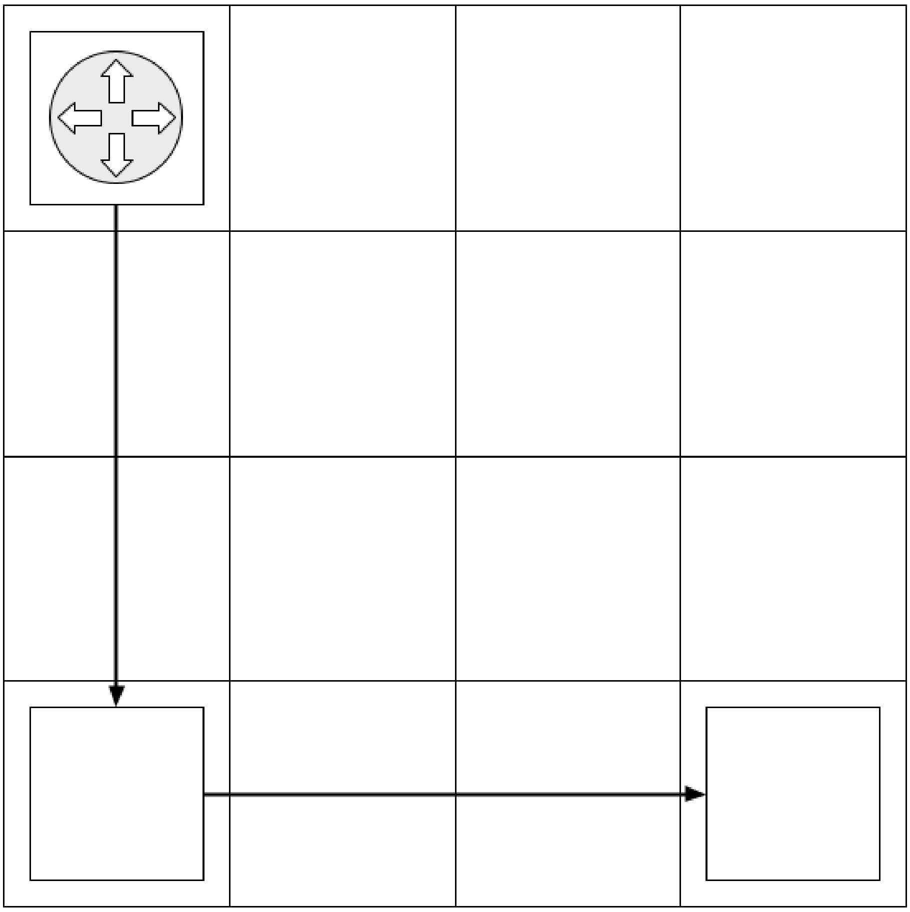
The second group prepared a square-shaped path that was used to introduce the new orange card. We imagined what the card would do, and then introduced it to the prepared path and observed how the robot responded to it (see Figure 11). The children were invited to reflect how to activate and deactivate the sound reproduction and to propose routes that integrate an orange card.

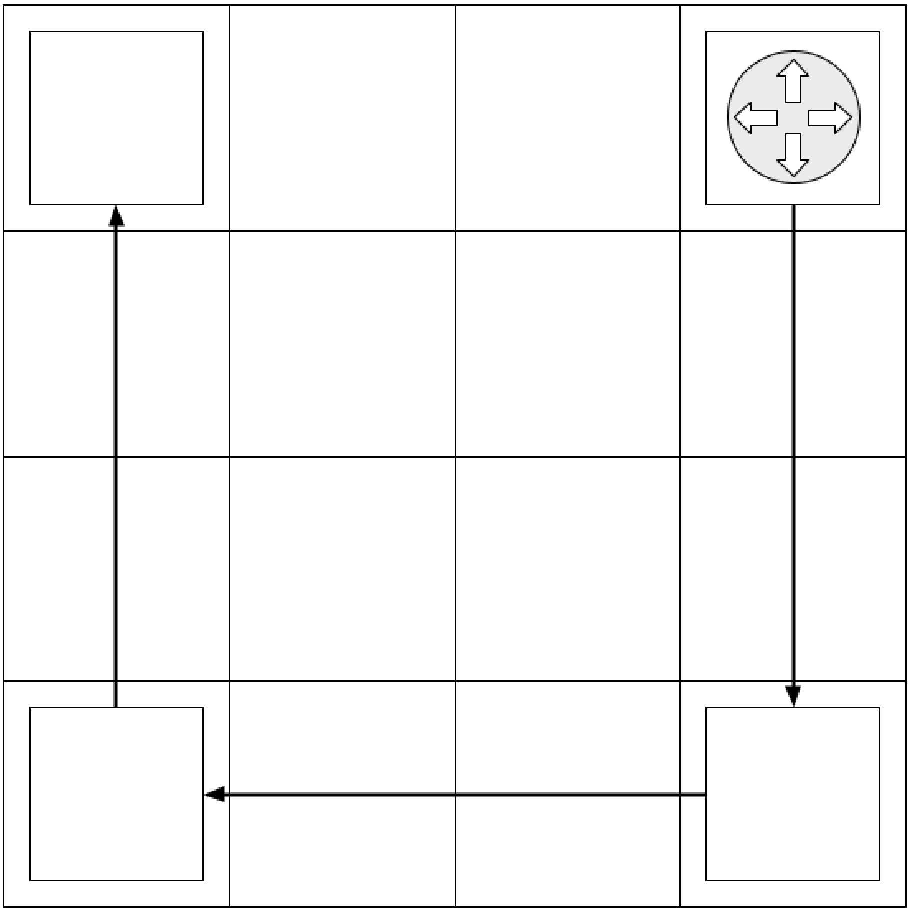
Activity #6 Total time: 60 minutes (10 minutes per group). We formed small groups (1 to 3 children) and each group worked with Robotito for about 10 minutes, while the rest of the class performed curricular activities with the teacher. The children in the small group were distributed around the 4 x 4 mat. Each child was invited to code with color cards a L-shaped path from a point next to it to one of the classmates or to the researcher. The initial orientation of the robot, the initial position, and the end point were defined by the researcher. In some cases, to make the task more challenging, we used white cards with an X in the middle that indicated that the robot should not pass through that cell. In other cases we asked the child to prepare the path and activate the music before arriving at the end point. All the group members participated in the final task in which they programmed a long path that incorporated music activation and deactivation.

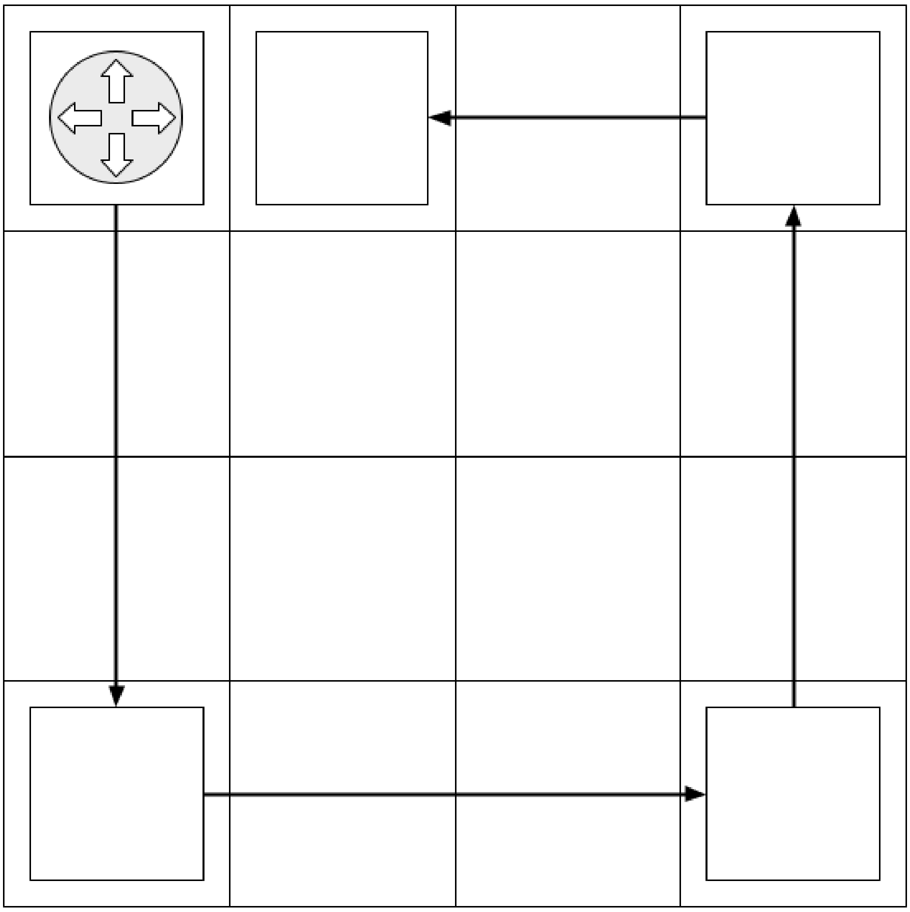
Activity #7 Total time: 30 minutes. The class was divided into two groups and each group worked with one researcher on the same task. First we introduced the pink coding card and thought how Robotito responds to it. After turning on the robot and observing how it acts. The researcher rotated the robot and each child tried to predict in which cell it would stop after sensing the pink card. We ended the session with building paths suggested by the children.

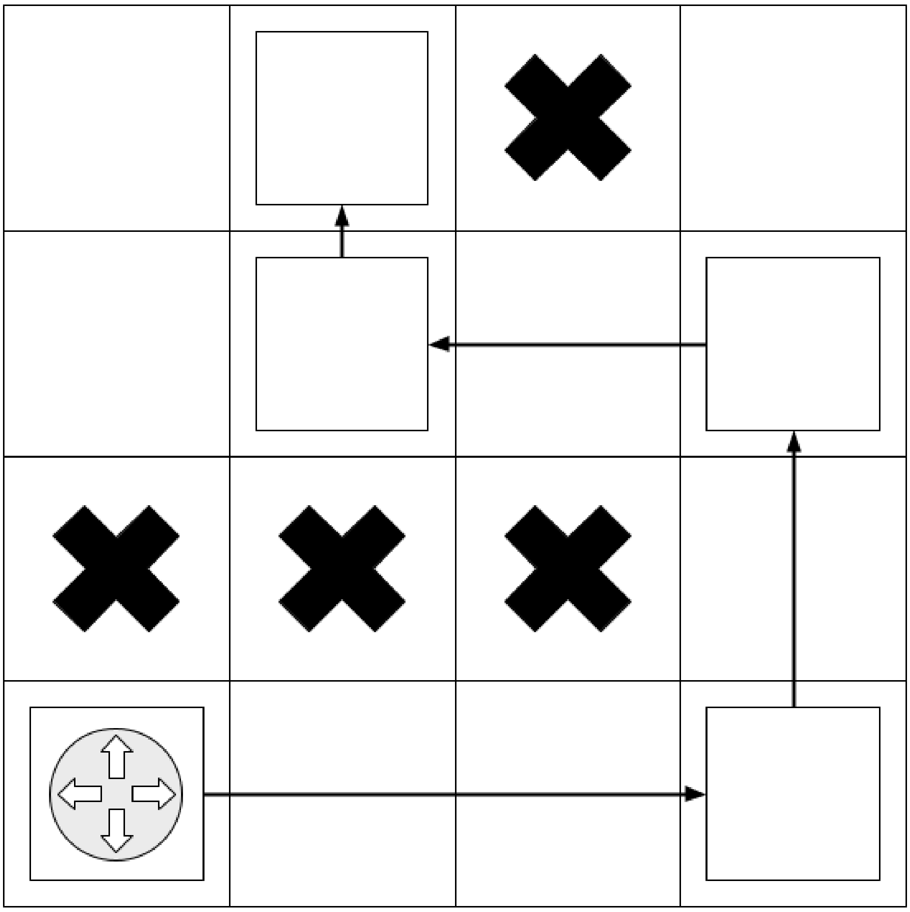
Activity #8 Total time: 75 minutes (10 to 15 per group). We formed small groups (1 to 3 children) and each group worked with Robotito for about 10 to 15 minutes, while the rest of the class performed curricular activities with the teacher. Each child had to solve a task based on combining the pink card (initial point) with other color cards to reach the purple card. In those exercises the rotation of the robot was fixed by the researcher. In the final exercise the children had to predict what happens when we build a diagonal with three pink cards that crosses the mat and end with the purple card.

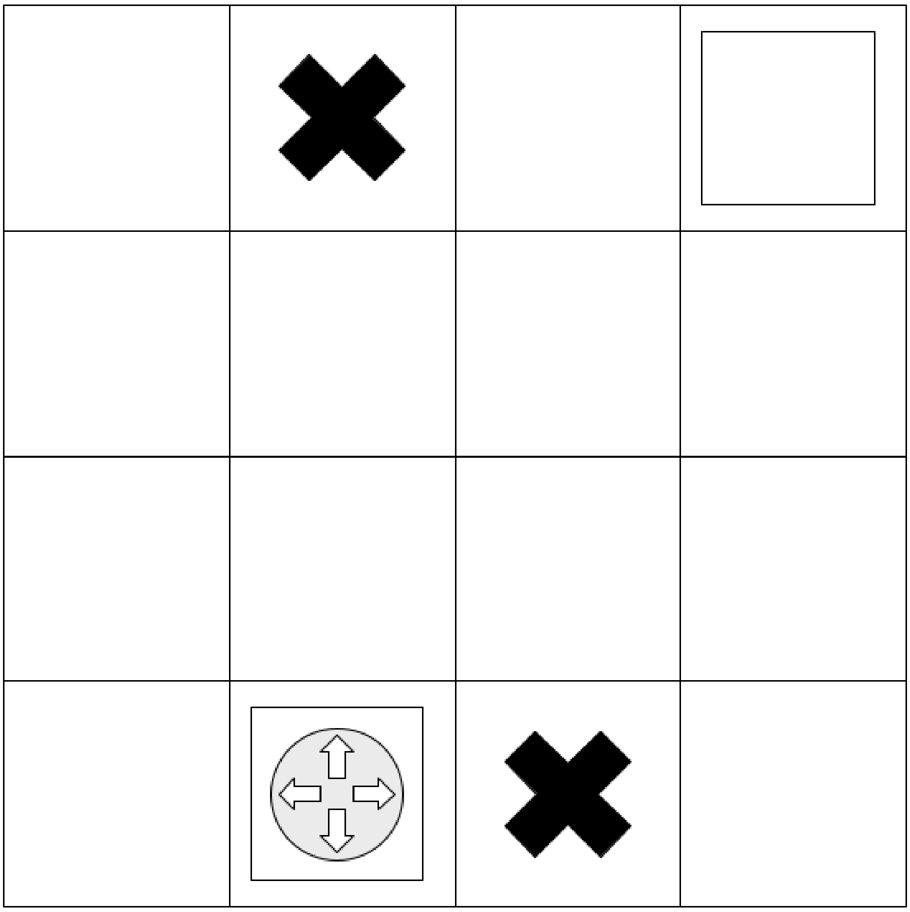
F On-paper tasks for activity #5

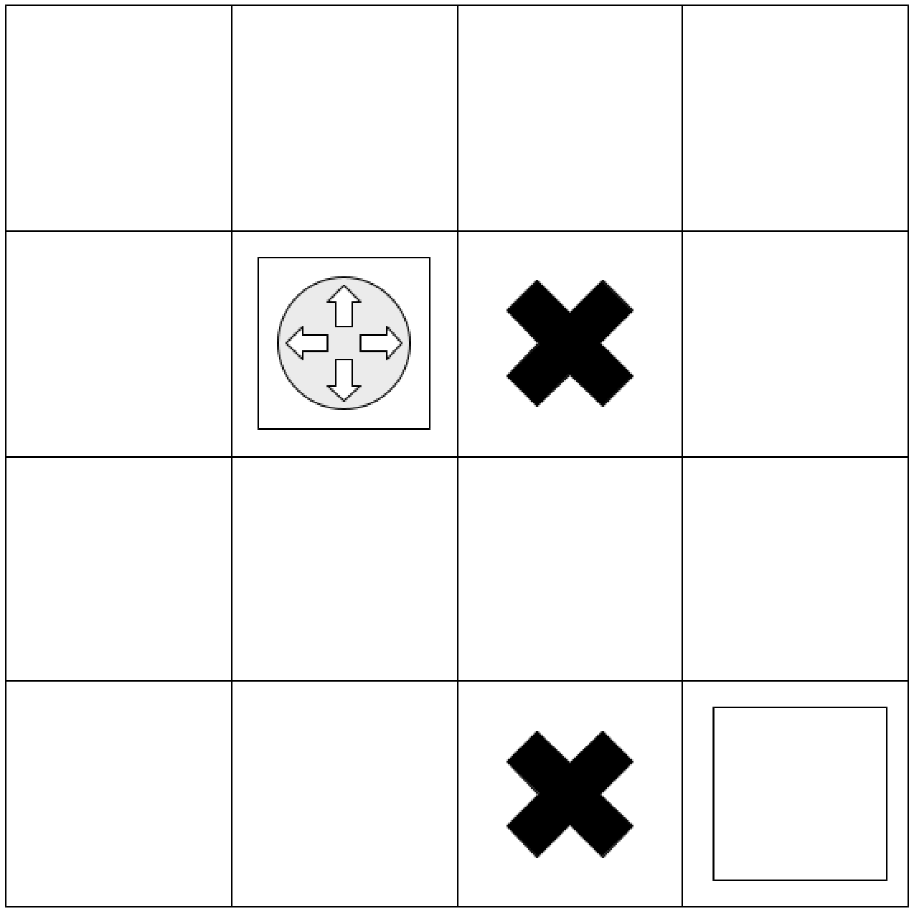


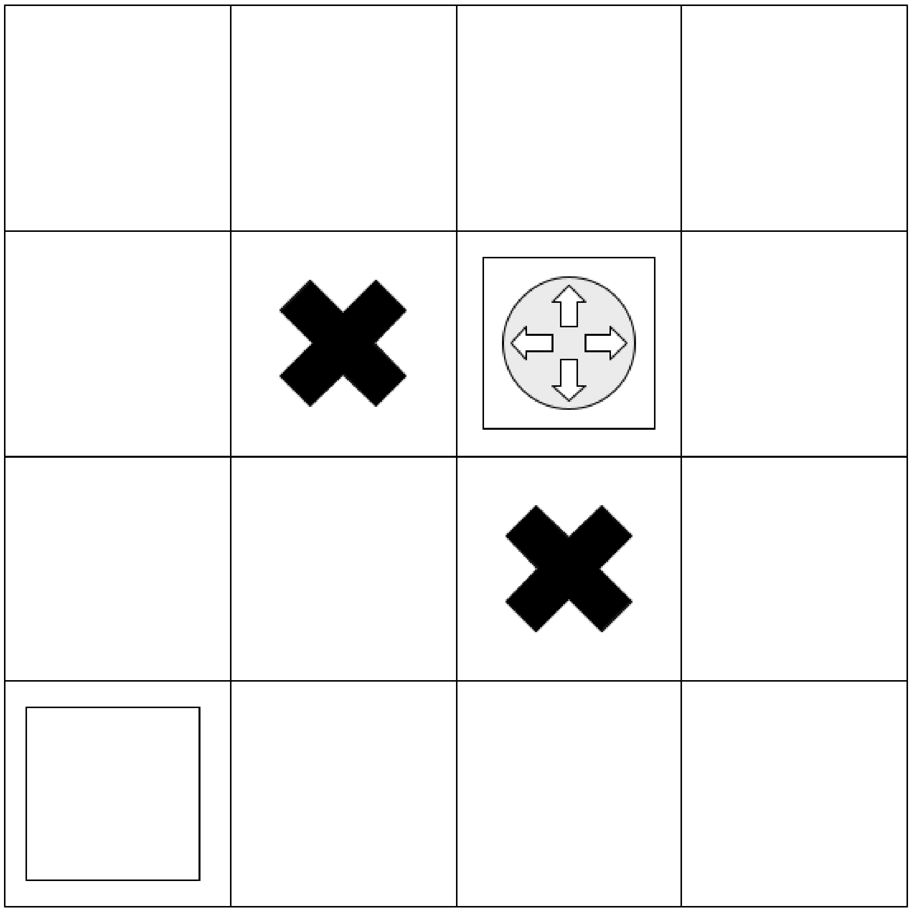












G BCTt results

ID	TYPE	GROUP	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
A2-1	PRE	A2	1	1	0	0	0	0	0	0	0	1	0	1	1
A2-2	PRE	A2	0	0	0	0	0	0	1	0	0	1	0	1	0
A2-3	PRE	A2	1	1	0	0	0	1	1	0	0	0	1	0	1
A2-4	PRE	A2	1	1	1	0	0	1	1	0	1	0	0	0	1
A2-5	PRE	A2	0	1	1	1	1	1	1	0	0	1	0	0	0
A2-6	PRE	A2	0	0	0	0	0	0	0	1	0	1	0	0	1
A2-7	PRE	A2	0	1	1	1	0	0	1	0	1	1	1	0	1
A2-8	PRE	A2	0	0	1	1	1	0	1	0	0	1	0	0	1
A2-9	PRE	A2	0	0	1	0	0	1	1	1	1	0	0	0	1
A2-10	PRE	A2	1	1	1	1	1	1	1	1	1	1	1	0	1
A2-11	PRE	A2	0	0	1	0	0	1	1	0	1	1	0	0	0
A2-12	PRE	A2	1	0	1	1	1	1	0	1	0	1	0	0	1
A2-13	PRE	A2	1	1	0	1	1	0	0	0	0	1	0	0	1
A2-14	PRE	A2	1	0	0	0	0	0	0	0	1	0	0	0	1
A2-15	PRE	A2	0	1	0	1	0	0	0	0	0	0	0	0	0
A2-16	PRE	A2	1	1	1	0	1	1	1	1	1	0	0	0	1
A2-17	PRE	A2	1	1	1	1	1	1	0	1	0	1	1	1	1
A2-18	PRE	A2	1	1	1	1	1	1	1	1	1	1	1	1	1
A2-19	PRE	A2	1	1	1	1	1	1	0	0	1	0	0	0	0
A1-1	PRE	A1	0	0	1	0	0	0	0	0	1	0	1	0	1
A1-2	PRE	A1	1	0	1	0	0	1	0	0	0	1	0	1	0
A1-3	PRE	A1	1	1	1	1	1	1	1	0	1	0	0	0	0
A1-4	PRE	A1	1	1	1	1	1	0	0	1	0	1	0	0	1
A1-5	PRE	A1	1	0	0	0	0	1	0	0	0	0	0	0	0
A1-6	PRE	A1	0	1	0	0	0	1	0	0	0	0	0	0	0
A1-7	PRE	A1	0	0	1	1	1	0	0	0	0	1	0	0	0
A1-8	PRE	A1	1	1	1	1	1	0	1	0	0	0	0	0	1
A1-9	PRE	A1	0	1	1	1	0	1	0	1	1	1	1	0	1
A1-10	PRE	A1	1	0	1	1	0	0	1	0	0	1	0	0	1
A1-11	PRE	A1	1	1	1	1	1	0	1	1	1	0	0	1	1
A1-12	PRE	A1	1	0	1	1	1	0	1	1	1	1	0	0	1
A1-13	PRE	A1	0	0	0	1	0	1	1	0	0	1	0	0	1
A1-14	PRE	A1	0	1	1	0	1	0	1	0	0	1	0	0	1
A1-15	PRE	A1	1	1	1	1	0	1	1	0	0	1	0	1	1
A1-16	PRE	A1	0	1	1	1	1	0	0	0	0	0	1	1	0
A1-17	PRE	A1	1	0	0	0	0	0	0	0	1	0	0	0	0
CG-1	PRE	CG	0	1	1	1	1	1	0	0	0	0	1	0	1
CG-2	PRE	CG	1	0	0	0	1	0	1	0	1	0	0	1	0
CG-3	PRE	CG	0	1	1	1	0	1	0	0	0	1	0	0	1
CG-4	PRE	CG	1	1	1	1	1	1	0	0	1	1	0	0	1
CG-5	PRE	CG	1	0	1	0	0	0	0	1	0	1	0	0	0
CG-6	PRE	CG	1	1	1	0	1	1	1	0	0	1	0	0	1
CG-7	PRE	CG	1	1	1	1	0	1	1	0	1	1	0	1	1
CG-8	PRE	CG	1	1	1	1	1	1	0	0	0	1	0	0	0
CG-9	PRE	CG	1	1	1	1	1	1	0	0	1	1	0	0	1
CG-10	PRE	CG	0	1	1	1	0	0	0	0	1	1	0	0	1
CG-11	PRE	CG	0	1	0	0	1	0	1	1	1	0	0	0	1
CG-12	PRE	CG	0	0	1	1	1	1	0	0	1	1	0	0	0
CG-13	PRE	CG	1	0	0	0	0	0	0	0	0	0	0	1	1
CG-14	PRE	CG	1	1	1	1	0	1	1	0	1	0	0	0	1
CG-15	PRE	CG	1	1	0	1	1	0	1	1	1	1	1	1	1
CG-16	PRE	CG	1	1	1	1	1	1	1	1	1	1	1	0	1
CG-17	PRE	CG	1	1	1	0	0	1	0	1	1	1	1	1	1
CG-18	PRE	CG	1	0	1	1	1	1	1	0	0	1	1	1	1
CG-19	PRE	CG	0	1	1	0	0	1	1	0	1	1	1	0	1
CG-20	PRE	CG	1	1	1	1	1	1	0	0	1	1	1	0	1
A2-1	POST	A2	1	1	1	1	1	0	0	0	1	1	0	0	1
A2-2	POST	A2	1	1	0	1	0	0	1	0	0	0	1	0	1
A2-3	POST	A2	0	0	1	1	0	0	0	0	0	0	0	0	0
A2-4	POST	A2	1	1	1	1	1	1	1	0	0	1	1	0	1
A2-5	POST	A2	1	1	1	1	1	1	1	0	0	1	0	1	1
A2-6	POST	A2	1	1	1	0	0	1	0	0	0	1	0	1	0
A2-7	POST	A2	1	1	1	1	1	1	1	1	1	1	0	1	1
A2-8	POST	A2	1	1	1	1	1	1	1	1	1	1	1	1	1
A2-9	POST	A2	1	0	1	0	0	0	1	0	1	1	0	0	1
A2-10	POST	A2	1	1	1	1	1	1	1	1	1	0	1	0	1
A2-11	POST	A2	1	1	1	1	1	1	1	0	1	1	1	0	1
A2-12	POST	A2	0	1	1	0	0	0	1	1	0	1	0	0	0
A2-13	POST	A2	1	1	1	0	0	0	1	0	1	0	1	0	0
A2-14	POST	A2	1	1	0	0	0	0	1	0	1	0	0	0	1
A2-15	POST	A2	0	1	0	0	0	0	0	0	1	0	0	1	0
A2-16	POST	A2	1	1	1	1	1	1	1	1	1	0	1	1	1
A2-17	POST	A2	1	1	1	1	1	0	0	1	1	1	1	1	1
A2-18	POST	A2	1	1	1	1	0	1	1	0	1	1	0	0	1
A2-19	POST	A2	1	1	1	1	1	1	1	0	1	1	0	0	1
A1-1	POST	A1	1	1	0	0	0	0	0	0	0	0	0	0	0
A1-2	POST	A1	0	0	1	1	1	0	1	1	0	1	1	0	0
A1-3	POST	A1	1	1	1	1	0	1	1	1	1	1	0	0	0
A1-4	POST	A1	1	1	1	1	0	0	0	0	1	1	1	0	0
A1-5	POST	A1	0	0	1	1	0	0	0	0	1	1	0	0	0
A1-6	POST	A1	1	1	1	1	1	0	0	0	1	1	1	0	0
A1-7	POST	A1	1	1	1	1	0	0	1	0	1	1	0	1	1
A1-8	POST	A1	0	0	0	0	0	0	1	0	1	0	1	1	1
A1-9	POST	A1	1	1	0	0	0	1	0	0	0	0	1	0	1
A1-10	POST	A1	1	0	0	0	1	0	1	0	0	0	1	0	1
A1-11	POST	A1	1	1	1	1	1	1	0	0	1	0	0	1	0
A1-12	POST	A1	1	1	1	1	1	0	1	0	1	1	0	0	1
A1-13	POST	A1	1	1	0	0	1	0	0	0	0	0	1	0	1
A1-14	POST	A1	1	1	1	1	0	0	0	0	1	1	0	0	1
A1-15	POST	A1	1	1	1	1	1	1	1	0	0	1	0	0	1
A1-16	POST	A1	0	1	0	1	0	1	0	0	1	1	0	0	1
A1-17	POST	A1	1	0	1	0	0	0	1	0	1	1	1	0	0
CG-1	POST	CG	1	0	0	0	1	0	1	0	1	0	1	0	1
CG-2	POST	CG	0	0	0	0	1	1	1	0	1	0	1	0	1
CG-3	POST	CG	0	1	1	1	0	1	0	0	1	0	0	0	1
CG-4	POST	CG	1	1	1	1	1	1	1	1	1	1	0	0	1
CG-5	POST	CG	1	1	1	0	0	1	1	0	0	1	0	0	1
CG-6	POST	CG	0	0	1	0	0	0	0	1	0	0	0	0	1
CG-7	POST	CG	1	1	1	1	1	1	1	0	0	1	1	0	1
CG-8	POST	CG	0	1	1	1	1	1	0	0	0	1	1	0	1
CG-9	POST	CG	1	1	1	1	1	1	1	1	1	1	1	1	1
CG-10	POST	CG	1	1	1	1	1	1	1	1	1	1	1	0	1
CG-11	POST	CG	1	0	0	1	0	0	1	0	1	0	0	0	1
CG-12	POST	CG	1	1	1	1	1	0	1	0	1	0	0	0	0
CG-13	POST	CG	0	0	1	1	0	1	0	0	0	0	0	1	0
CG-14	POST	CG	1	1	1	1	1	0	1	1	1	1	1	0	1
CG-15	POST	CG	1	1	1	1	1	1	1	0	1	0	0	1	1
CG-16	POST	CG	1	1	1	1	1	1	1	1	1	1	1	0	1
CG-17	POST	CG	1	1	1	1	1	1	1	1	1	1	0	0	1
CG-18	POST	CG	1	1	1	1	1	0	1	0	0	0	1	1	1
CG-19	POST	CG	1	1	1	1	1	1	1	0	1	1	1	0	1
CG-20	POST	CG	1	1	1	1	1	1	1	0	1	1	1	0	1

H TechCheck-K results

ID	TYPE	GROUP	P1	P2	P3	P4	P5	P6	P7	P8	P9
A2-1	PRE	A2	0	0	1	1	1	1	0	1	1
A2-2	PRE	A2	1	0	0	1	0	0	0	1	1
A2-3	PRE	A2	0	0	0	1	1	1	0	1	0
A2-4	PRE	A2	0	1	1	0	1	0	1	1	1
A2-5	PRE	A2	0	1	0	0	0	0	0	1	1
A2-6	PRE	A2	1	1	1	0	0	0	0	1	1
A2-7	PRE	A2	1	1	1	0	0	0	1	1	1
A2-8	PRE	A2	1	0	1	1	0	0	0	1	1
A2-9	PRE	A2	0	1	1	1	0	1	1	1	1
A2-10	PRE	A2	1	0	0	1	0	0	1	1	1
A2-11	PRE	A2	1	0	1	1	0	0	0	1	1
A2-12	PRE	A2	0	0	0	1	0	0	1	1	1
A2-13	PRE	A2	0	1	0	1	0	1	0	1	0
A2-14	PRE	A2	0	1	1	0	1	1	0	0	1
A2-15	PRE	A2	1	0	1	0	0	0	0	0	0
A2-16	PRE	A2	0	1	1	1	0	0	0	1	1
A2-17	PRE	A2	1	1	0	0	0	0	1	1	1
A2-18	PRE	A2	1	1	1	0	0	1	0	1	0
A2-19	PRE	A2	1	1	0	1	1	0	0	1	1
A1-1	PRE	A1	0	1	1	0	0	0	0	0	1
A1-2	PRE	A1	1	1	1	0	1	1	0	1	1
A1-3	PRE	A1	0	0	0	1	0	0	0	1	1
A1-4	PRE	A1	1	1	0	0	0	0	1	1	1
A1-5	PRE	A1	1	1	1	0	0	1	0	1	1
A1-6	PRE	A1	0	1	0	1	0	0	0	0	0
A1-7	PRE	A1	1	1	0	0	1	0	0	1	1
A1-8	PRE	A1	1	1	1	1	0	0	1	1	1
A1-9	PRE	A1	0	1	0	0	1	0	1	1	1
A1-10	PRE	A1	1	1	0	1	0	1	1	1	1
A1-11	PRE	A1	0	1	1	0	1	0	0	1	1
A1-12	PRE	A1	0	1	1	0	1	0	0	0	0
A1-13	PRE	A1	0	0	1	0	0	0	1	0	0
A1-14	PRE	A1	0	1	1	0	0	0	1	1	0
A1-15	PRE	A1	0	1	1	0	0	0	1	1	1
A1-16	PRE	A1	0	0	0	1	0	1	0	1	0
A1-17	PRE	A1	1	1	0	0	0	1	0	0	0
CG-1	PRE	CG	0	1	1	0	0	0	0	1	1
CG-2	PRE	CG	1	0	0	0	1	0	0	0	1
CG-3	PRE	CG	0	0	1	1	1	0	0	1	0
CG-4	PRE	CG	0	1	0	0	1	0	1	1	1
CG-5	PRE	CG	0	0	1	0	0	0	1	1	0
CG-6	PRE	CG	0	1	0	0	0	0	1	1	1
CG-7	PRE	CG	1	1	1	0	0	0	0	1	1
CG-8	PRE	CG	1	1	1	1	0	0	1	1	0
CG-9	PRE	CG	0	1	0	1	0	0	0	1	1
CG-10	PRE	CG	1	0	0	1	0	1	0	1	1
CG-11	PRE	CG	0	1	1	0	0	0	1	1	0
CG-12	PRE	CG	1	1	1	0	0	0	0	1	1
CG-13	PRE	CG	0	0	1	0	0	0	1	0	1
CG-14	PRE	CG	1	1	1	0	1	0	1	1	0
CG-15	PRE	CG	1	1	1	0	0	1	1	1	1
CG-16	PRE	CG	0	1	1	0	0	0	0	1	1
CG-17	PRE	CG	1	0	0	1	0	1	1	1	1
CG-18	PRE	CG	1	1	1	1	0	1	1	1	1
CG-19	PRE	CG	0	0	0	0	0	0	0	1	1
CG-20	PRE	CG	1	1	0	0	1	0	0	1	1
A2-1	POST	A2	1	1	1	0	0	1	0	1	1
A2-2	POST	A2	0	0	1	1	0	0	0	1	1
A2-3	POST	A2	1	0	0	0	0	0	0	1	1
A2-4	POST	A2	1	1	0	1	0	0	0	1	1
A2-5	POST	A2	1	1	1	1	1	1	0	1	1
A2-6	POST	A2	1	1	0	0	0	0	0	1	1
A2-7	POST	A2	1	0	0	1	1	0	1	1	1
A2-8	POST	A2	1	1	1	0	0	1	1	1	1
A2-9	POST	A2	1	0	0	0	0	1	0	1	1
A2-10	POST	A2	1	1	1	1	1	0	0	1	1
A2-11	POST	A2	1	0	1	0	0	0	1	1	1
A2-12	POST	A2	1	1	0	0	0	1	0	1	1
A2-13	POST	A2	0	1	1	0	0	0	0	1	1
A2-14	POST	A2	1	1	1	0	1	0	0	1	1
A2-15	POST	A2	1	0	0	0	0	1	0	0	0
A2-16	POST	A2	1	1	0	1	0	0	0	1	1
A2-17	POST	A2	1	1	0	1	0	0	1	1	1
A2-18	POST	A2	1	1	0	1	1	1	1	1	1
A2-19	POST	A2	1	1	1	1	1	0	0	1	1
A1-1	POST	A1	1	0	0	0	0	0	1	1	1
A1-2	POST	A1	0	0	0	0	1	1	0	1	1
A1-3	POST	A1	0	1	0	1	1	1	1	1	1
A1-4	POST	A1	1	1	1	0	0	1	1	1	1
A1-5	POST	A1	1	1	0	0	0	1	0	1	1
A1-6	POST	A1	0	1	0	0	0	1	0	1	1
A1-7	POST	A1	0	1	1	0	0	0	0	1	0
A1-8	POST	A1	1	1	0	1	0	0	0	1	1
A1-9	POST	A1	1	1	1	0	0	0	0	0	0
A1-10	POST	A1	0	1	1	1	0	0	0	1	1
A1-11	POST	A1	0	0	1	1	0	1	1	1	1
A1-12	POST	A1	1	1	0	1	0	0	1	1	1
A1-13	POST	A1	0	1	1	0	1	0	0	1	1
A1-14	POST	A1	1	1	1	0	0	0	0	1	1
A1-15	POST	A1	1	1	1	1	1	1	0	1	1
A1-16	POST	A1	1	1	1	0	0	0	0	1	1
A1-17	POST	A1	0	1	1	1	0	1	0	1	1
CG-1	POST	CG	0	0	0	0	0	0	0	1	1
CG-2	POST	CG	0	0	1	0	0	0	0	1	1
CG-3	POST	CG	1	0	1	0	1	1	0	1	1
CG-4	POST	CG	1	1	1	1	1	1	0	1	1
CG-5	POST	CG	0	0	0	0	0	0	0	1	0
CG-6	POST	CG	1	1	1	0	0	0	0	1	1
CG-7	POST	CG	0	0	0	1	1	0	1	1	1
CG-8	POST	CG	1	1	1	1	1	0	0	1	1
CG-9	POST	CG	0	1	0	1	0	1	1	1	1
CG-10	POST	CG	1	0	0	1	0	0	0	1	1
CG-11	POST	CG	0	1	1	0	0	0	0	1	1
CG-12	POST	CG	1	1	1	0	0	0	0	1	1
CG-13	POST	CG	0	1	1	0	0	1	0	1	1
CG-14	POST	CG	1	1	1	1	1	1	1	1	1
CG-15	POST	CG	0	1	1	1	0	1	0	0	1
CG-16	POST	CG	1	0	1	1	0	0	1	1	1
CG-17	POST	CG	1	0	1	1	1	0	1	1	1
CG-18	POST	CG	0	0	0	1	0	0	0	1	1
CG-19	POST	CG	0	0	1	0	0	1	0	1	1
CG-20	POST	CG	1	1	0	0	1	0	0	1	1

I Robotito Test results

ID	GROUP	P1	P2	P3	P4
A2-1	A2	1	1	1	1
A2-2	A2	1	1	0	1
A2-3	A2	1	0	0	1
A2-4	A2	1	1	1	1
A2-5	A2	1	1	0	1
A2-6	A2	1	0	1	1
A2-7	A2	1	1	1	1
A2-8	A2	1	1	0	1
A2-9	A2	1	1	1	0
A2-10	A2	1	1	1	1
A2-11	A2	1	1	1	1
A2-12	A2	1	1	0	0
A2-13	A2	1	1	0	0
A2-14	A2	1	0	0	0
A2-15	A2	1	0	0	0
A2-16	A2	1	1	0	1
A2-17	A2	1	1	1	1
A2-18	A2	1	1	1	1
A2-19	A2	1	1	0	1
A1-1	A1	1	1	0	1
A1-2	A1	1	0	0	0
A1-3	A1	1	1	1	1
A1-4	A1	1	1	1	1
A1-5	A1	1	1	1	1
A1-6	A1	1	1	1	0
A1-7	A1	1	1	1	1
A1-8	A1	1	0	1	0
A1-9	A1	1	1	1	0
A1-10	A1	1	1	0	0
A1-11	A1	1	1	1	1
A1-12	A1	1	1	0	0
A1-13	A1	1	0	0	0
A1-14	A1	1	1	0	1
A1-15	A1	1	1	1	1
A1-16	A1	1	0	1	0
A1-17	A1	1	1	1	0

Appendix 7

Authorship Statements of the Co-authors

Iowa City, 12 de septiembre de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal de los siguientes artículos, de los que soy coautor, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 29, 100337.

Bakala, E., Gerosa, A., Hourcade, J. P., Pascale, M., Hergatacorzian, C., & Tejera, G. (2022). Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. *In Proceedings of the 21st annual ACM Interaction Design and Children Conference* (pp. 422-429).

Bakala, E., Gerosa, A., Hourcade, J. P., Tejera, G., Peterman, K., & Trinidad, G. (2022). A systematic review of technologies to teach control structures in preschool education. *Frontiers in Psychology*, 13, 911057.

Bakala, E., Pires, A. C., Tejera, G., & Hourcade, J. P. (2023). “It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers. In *Proceedings of the 2023 ACM Conference on Information Technology for Social Good* (pp. 477-486).

Bakala, E., Tejera, G., & Hourcade, J.P. (2024). An iterative design and empirical evaluation of conditionals for Robotito. Tech. rep. Udelar. FI., 2024. url: https://hdl.handle.net/20.500.12008/45833

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

Dr. Juan Pablo Hourcade

Montevideo, 12 de septiembre de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal de los siguientes artículos, de los que soy coautor, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 29, 100337.

Bakala, E., Gerosa, A., Hourcade, J. P., Pascale, M., Hergatacorzian, C., & Tejera, G. (2022). Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. *In Proceedings of the 21st annual ACM Interaction Design and Children Conference* (pp. 422-429).

Bakala, E., Gerosa, A., Hourcade, J. P., Tejera, G., Peterman, K., & Trinidad, G. (2022). A systematic review of technologies to teach control structures in preschool education. *Frontiers in Psychology*, 13, 911057.

Bakala, E., Pires, A. C., Tejera, G., & Hourcade, J. P. (2023). “It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers. In *Proceedings of the 2023 ACM Conference on Information Technology for Social Good* (pp. 477-486).

Bakala, E., Tejera, G., & Hourcade, J. P. (2024). An iterative design and empirical evaluation of conditionals for Robotito. Tech. rep. Udelar. FI., 2024. url: https://hdl.handle.net/20.500.12008/45833

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

Dr. Gonzalo Tejera

Lisboa, 17 de junio de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal del siguiente artículo, del que soy coautora, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Pires, A. C., Tejera, G., & Hourcade, J. P. (2023). “It will surely fall”: Exploring Teachers’ Perspectives on Commercial Robots for Preschoolers. In *Proceedings of the 2023 ACM Conference on Information Technology for Social Good* (pp. 477-486).

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

Dr. Ana Cristina Pires

Helsinki, 17 de junio de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal del siguiente artículo, del que soy coautora, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., Pascale, M., Hergatacorzian, C., & Tejera, G. (2022). Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. *In Proceedings of the 21st annual ACM Interaction Design and Children Conference* (pp. 422-429).

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

Mag. Camila Hergatacorzian

Montevideo, 17 de junio de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal del siguiente artículo, del que soy coautor, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., Tejera, G., Peterman, K., & Trinidad, G. (2022). A systematic review of technologies to teach control structures in preschool education. *Frontiers in Psychology*, 13, 911057.

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

GUILLERMO TRINIDAD
Mag. Guillermo Trinidad

Montevideo, 17 de junio de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal del siguiente artículo, del que soy coautora, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., Pascale, M., Hergatacorzian, C., & Tejera, G. (2022). Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. *In Proceedings of the 21st annual ACM Interaction Design and Children Conference* (pp. 422-429).

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

M. Pascale

Mag. María Pascale

Iowa City, 17 of June 2024

Comisión de posgrado,
Área Informática
PEDECIBA

To Whom It May Concern:

I hereby declare that Ewelina Bakala has been the principal author of the following article, of which I am a co-author, and that I have no objection to its inclusion in her *Thesis by Compilation of Articles*.

Bakala, E., Gerosa, A., Hourcade, J. P., Tejera, G., Peterman, K., & Trinidad, G. (2022). A systematic review of technologies to teach control structures in preschool education. *Frontiers in Psychology,* 13, 911057.

Additionally, I declare that I have not included nor will I include said article in a thesis of the compilation of articles type of my own authorship.

Sincerely,

Kerry Peterman

Montevideo, 17 de junio de 2024

Comisión de posgrado,
Área Informática
PEDECIBA

De mi consideración:

Por la presente declaro que Ewelina Bakala ha sido la autora principal de los siguientes artículos, de los que soy coautora, y que no tengo ninguna objeción a su inclusión en su *Tesis por Compilación de Artículos.*

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 29, 100337.

Bakala, E., Gerosa, A., Hourcade, J. P., Pascale, M., Hergatacorzian, C., & Tejera, G. (2022). Design factors affecting the social use of programmable robots to learn computational thinking in kindergarten. *In Proceedings of the 21st annual ACM Interaction Design and Children Conference* (pp. 422-429).

Bakala, E., Gerosa, A., Hourcade, J. P., Tejera, G., Peterman, K., & Trinidad, G. (2022). A systematic review of technologies to teach control structures in preschool education. *Frontiers in Psychology*, 13, 911057.

Asimismo, declaro que no he incluido ni incluiré dicho artículo en una tesis del tipo de compilación de artículos de mi autoría.

Saluda Atentamente,

Mag. Anaclara Gerosa