Early Childhood Education Journal
https://doi.org/10.1007/s10643-023-01563-y

Investigating Children's Programming Skills Through Play with Robots (KIBO)

Joohi Lee[1] · Sham'ah Yunus[2] · Joo Ok Lee[3]

Accepted: 9 August 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract
Robotics has emerged as a popular interdisciplinary pedagogical approach in the field of education to teach children STEM concepts. By providing playful learning experiences, the use of robots engages children in an active learning process, making it an effective tool to promote their targeted knowledge, skills, and disposition towards STEM, including programming and coding skills. This study aimed to investigate the impact of using a programmable robot (KIBO) as a station activity and how it impacts pre-kindergarteners' programming skills after 2 weeks of play. Fourteen children aged 3–4 participated in the study during the summer, with the robot station activity lasting for 40 min each day. Results indicated a significant improvement in all areas of programming skills, including copying programming (n = 13), programming (n = 13), programming with a conditional statement (n = 7), and decoding (n = 10), after playing with the robot. The findings suggest that incorporating programmable robots as station activities can be a valuable addition to pre-kindergarten curriculums to promote basic programming skills in young children, which can serve as a foundation for future STEM learning. Further research is needed to determine the long-term effects of using robots as a teaching tool for young children.

Keywords KIBO · Copying programming · Programming · Programming with a condition statement · Decoding

Educational robotics, also referred to as pedagogical robotics, is a dynamic subfield of robotics that aims to provide children with interactive and interdisciplinary experiences involving robotics and programming (Govind & Bers, 2021). By engaging with educational robots, children can develop essential thinking skills, including problem-solving (Chevalier et al., 2020) and creativity (Eguchi, 2014; Romero et al., 2017). Moreover, programming activities nurture children's computational thinking skills, which are increasingly crucial in the twenty-first century (International Society for Technology Education & Computer Science Teachers Association, 2011; Lee & Junoh, 2019).

The focus of robotics in education has evolved from a technical-centric approach, solely focused on aspects like robot development, design, and building (Alimisis, 2012), to a more interdisciplinary perspective that integrates robotics into the learning process across disciplines.

In the current robotics pedagogy, learners are actively engaged in hands-on experiences to enhance their thinking skills in diverse fields (Di Lieto et al., 2016). Educational robots are widely recognized as effective tools for creating engaging and creative learning experiences for children (Eguchi, 2014; Cherniak et al., 2019; Romero et al., 2017), as well as for advancing pedagogical knowledge among teachers (Govind & Bers, 2021).

Despite the growing body of studies investigating the effectiveness of using educational robots in interdisciplinary settings for young children (Chevalier et al., 2020; Govind & Bers, 2021), there is still a need for further research in this area (Di Lieto et al., 2016). Expanding research in this field will deepen our understanding of the benefits and implications of incorporating educational robotics into

✉ Joohi Lee
joohilee@uta.edu

Sham'ah Yunus
smdyunus@eiu.edu

Joo Ok Lee
hadasa5@hanmail.net

1 Early Childhood Math Education, Curriculum and Instruction, College of Education, University of Texas at Arlington, Arlington, TX, USA

2 Early Childhood Education, Eastern Illinois University, Charleston, IL, USA

3 Early Childhood Education, Busan National University of Education, Busan, South Korea

Published online: 01 September 2023

diverse educational contexts, ensuring that educators and policymakers can make informed decisions regarding its implementation.

In this study, we aimed to explore pre-kindergarteners' programming skills and the utilization of robots as a station activity. Station activities during free play time are common in early childhood education and follow a child-centered approach, where teachers facilitate learning by providing materials in each station based on children's interests (NAEYC, n.d.). For this study, a robotics station was introduced to the existing stations in a pre-k classroom, allowing children to select and rotate between stations during free play time. By integrating robots as part of the station activities, we sought to examine the impact on children's programming skills, creativity, and engagement (Eguchi, 2014; Romero et al., 2017). Additionally, we aimed to gain insights into the practical implementation of educational robotics in early childhood settings, thereby contributing to the existing body of knowledge in this field (Cherniak et al., 2019; Govind & Bers, 2021). Through this study, we aimed to provide valuable empirical evidence to inform educators and policymakers about the potential benefits and challenges of incorporating robotics into pre-kindergarten education, ultimately supporting the development of effective and evidence-based educational practices.

Robotics in Early Childhood

Emerging research in the field of early childhood education suggests that incorporating robotics into the curriculum can engage children's learning and enhance their computer science knowledge and skills (Lee, 2020; Lee et al., 2023; Sullivan & Bers, 2016; Gerosa et al., 2022).

Di Lieto et al. (2016) utilized a robotics curriculum system with Beebot and integrated it into early childhood classrooms through curricular units that combined programming and engineering with other subjects. Their study revealed that 5–6-year old children showed "a significant improvement in visual-spatial working memory and inhibition skills after using BeeBot" (p. 21).

Similarly, Bers (2010) found that children aged 5–7 years could effectively use the robot KIWI (Kids Invent with Imagination) to increase their STEM knowledge and skills. In her study, children learned basic engineering, mathematical, and programming concepts by creating, playing with, and decorating robots using gears, levers, motors, and sensors. Sullivan and Bers (2016) explored the effectiveness of the "Playmaker Program," a robotics education program in early childhood that used various screen-free robots, including KIBO, to teach programming, coding, and robotics in preschool settings. The study revealed that engaging with KIBO fostered children's interest in robotics and significantly enhanced their programming skills.

Young children have shown a natural inclination to be drawn to robots, perceiving them as appealing toys. This inherent attraction makes robots an effective and engaging tool for introducing robotics education to young learners. Studies by Keren and Fridin (2014) and Vizner and Strawhacker (2016) have highlighted the positive impact of using humanoid robots, like Nao, on children's programming and communication skills. The introduction of robots into the educational setting has been found to enhance children's abilities in these areas, contributing to their overall cognitive development.

Furthermore, Vizner and Strawhacker's research indicated that children of the same age demonstrated varying levels of programming behaviors and engaged in different hierarchical levels during programming tasks. This observation underscores the importance of personalized and adaptive learning approaches when incorporating robots into early childhood education.

Research on technology interventions integrating robots into early childhood education has also highlighted the potential benefits of early exposure to STEM (Science, Technology, Engineering, and Mathematics) curriculum and programming. Bers (2012) found that integrating robots in educational settings can foster equitable access to STEM knowledge and encourage children from diverse backgrounds to pursue STEM careers in the future. By leveraging the appeal of robots and creating a playful, collaborative learning environment, early education programs can effectively introduce fundamental technology and engineering concepts to young children during their formative years (Bers, 2018a).

In sum, robots offer a promising avenue for enhancing early childhood education in the domain of STEM learning (Bers, 2018b). Their intrinsic appeal and interactive nature facilitate children's engagement and proficiency in programming and communication skills. By incorporating robots into early education, educators can pave the way for a more inclusive and empowering approach to STEM education, nurturing the interests and abilities of young learners from diverse backgrounds (Sullivan & Bers, 2016). As such, integrating robotics into early childhood education can play a pivotal role in preparing the next generation for future challenges in technology and engineering fields.

Programming in Early Childhood

Robotics is an engaging activity that often requires programming skills unless the robot is pre-programmed. Programming or coding is not just a technical skill, but rather a crucial way of thinking to achieve literacy in the twenty-first century, similar to reading and writing. By learning to program or code, children engage in new ways of thinking that are often referred to as computational thinking (CT; International Society for Technology Education [ISTE] and Computer Science Teachers Association [CSTA], 2011). Programming, or coding, is a well-known tool to promote children's computational thinking that can help children engage in various computer science thinking skills such as logical and abstract thinking, problem-solving, and the creative design process (Flannery & Bers, 2013). According to Wing (2011), CT involves a range of analytical mental tools inherent to the field of computer science, such as thinking recursively, applying abstraction, ignoring irrelevant tasks when figuring out complex tasks, and using heuristic reasoning to discover solutions. The ISTE and CSTA strongly recommend that all students learn to program and code to develop their CT skills and become efficient problem solvers in a complex digital society.

Programming often involves writing codes in specific computer languages. Traditionally, these languages have been taught in middle and high schools using abstract forms of language such as Java, Python, and C++, among others. However, with the recognition of programming and coding as essential skills, computer languages have been modified for younger children using concrete representations such as blocks and icons. Several studies have investigated the effectiveness of block-based programming languages that remove or limit abstract texts in early childhood robotics (Govind & Bers, 2021; Strawhacker & Bers, 2019).

Block-based programming languages have gained significant popularity in early and elementary education due to their effectiveness in promoting basic programming skills and critical thinking among young learners, all without the need for reading and writing (Author, 2020). There are two types of block-based programming: unplugged and plugged. Unplugged programming does not require a computer and utilizes the same block-based languages. Visual aids, such as directional signs, command blocks, and directional arrows, are used to facilitate the programming process. On the other hand, block-based plugged programming involves computer involvement (Lee, 2020). For instance, robotic systems like KIBO utilize tangible programming blocks with an integrated computer chip, providing a more hands-on and concrete approach to programming without the need for an actual computer screen. Considering children's developmental level, this tactile interaction with tangible blocks enhances the learning experience, especially for younger children. In contrast, platforms like ScratchJr and Code.org rely on virtual block-based programming through the use of a computer screen. LEGO, depending on the purpose, employs a combination of both screen and non-screen elements in its block-based programming activities.

Studies have shown that children, as young as 3 years of age, can program using board game-like tools without involving computers. For example, Kazakoff et al. (2013) used a "Robot Turtles" board game to teach programming and coding concepts to children aged three and up. The game utilized a programmable robot within a context where turtles had to find jewels using directional commands such as "turn right," "turn left," and "move forward" to program their turtles to reach the jewels. The study revealed that this game effectively engaged young children in developing CT skills by creating sequences and solving problems.

According to Sullivan and Bers (2019), in addition to programming skills, children's literacy skills also improve as they participate in playing with a robot, and both problem-solving and personal expression skills emerge. Additionally, programming and coding can be a means of self-expression, similar to writing, speaking, and the arts, as children use programming and coding to communicate with a robot and to enable a robot to express the commands. However, Taylor and his research team (2017) suggest that the educational benefits of programming and coding, such as math skills and problem-solving, may not be as evident in early childhood as in other learning domains, such as verbal, causal, and social reasoning. These findings suggest that there is a developmental progression to the way children learn programming knowledge, which aligns with Piaget's (1953) argument that children's diverse responses are indicative of their different points along their general intellectual developmental trajectory.

The research on programming and coding for children has significant implications for the importance of introducing these concepts at a young age. Studies have demonstrated that children exhibit a range of mastery and creativity through coding but showed a complete lack of understanding about the most foundational programming concepts after the intervention (Di Lieto et al., 2016). This supports the growing body of research about the benefits of computer programming in supporting the development of other cognitive domains (Relkin et al., 2021). Programming is rapidly becoming a new literacy that children can use to connect with the logic of algorithms or the expressive possibilities of digital storytelling (Bers, 2018b). Current approaches to early childhood computer science education can be broadly categorized into four aspects: "unplugged computer science, block-based programming languages, programming games, and introductory robotic systems" (Kazakoff et al., 2013, p. 246). All of these approaches expand children's knowledge and skills of computer science concepts and skills, which include topics such as hardware and software, inputs, outputs, and other programming concepts (Kazakoff et al., 2013).

As discussed earlier, block-based programming has been widely employed in early childhood education, enabling children to engage in concrete block coding activities.

For example, in a block-based web programming tool such as Scratch or ScratchJr, children move command blocks with instructions to a workspace to program an object to complete an action based on the command. KIBO, a tangible robot that integrates concrete blocks, is also block-based. Studies focusing on block-based programming have revealed various positive impacts on promoting children's programming, coding, algorithmic, and creative thinking skills (Govind & Bers, 2021; Lott et al., 2019; Relkin et al., 2021; Strawhacker & Bers, 2019). Research has shown that even children as young as 3 years of age can effectively engage with a codable robot and develop foundational coding and computational skills (Govind & Bers, 2021).

In this study, the researchers utilized KIBO, a block-based programming robot, as a tool for a robotics station where children can engage in daily play based on their individual interests. The primary objective of the study is to examine children's programming skills both before and after the implementation of the robotics station. Specifically, the researchers assessed children's programming skills in four areas: programming, programming copying, programming with conditional statements, and decoding. The study employed specific operational definitions for each of these skills, which are detailed below.

Operational Definitions

The following operational definitions are used in this study.

- Programming: the process of creating a set of instructions or commands how a robot perform a task.
- Programming Copy: the process of copying a program.
- Programming with a Condition Statement: programming follows a condition command. In this study, “Wait for Clap” is used as the condition statement.
- Decoding: the process of interpreting programming/coding based on the movement of the robot.

Methodology

Participants and Context

Fourteen pre-kindergarten children aged 3-to 4-years old participated in this 2-week study, including six boys and eight girls with diverse racial and ethnic backgrounds (7 African American, 6 Latinx, and 1 Caucasian). The study took place in a non-profit childcare center located in an urban area in a southern state, following the Texas Education Agency's preschool enrollment guideline (TEA, 2021) and prioritizing enrollment of low-income families, English language learners, and children with special needs. During the summer program, a half-day program was available, and teacher-selected activities were implemented. For this study, a robotics station was integrated with three sets of KIBO for children to play during station time Children rotated through seven stations, including three required stations (reading, puzzle/math, and robotics) and four optional stations (art, dramatic, block, and computer). Free play time lasted about 40 min per day. During free play time, at least one adult (researcher) stayed in the station and assisted children when needed. One head teacher and one assistant teacher monitored children while walking around the stations and facilitated (e.g., asking questions, providing comments/suggestions, or complimenting, etc.) or observed children’s play.

Parent Consent and Children’s Assent

Before implementing the KIBO robotics station, a flyer for parent consent form was sent to 15 parents, and 14 parents consented for their children to participate the project. Researchers didn’t collect the data from the child whose parents didn’t send the consent form back. Researchers also collected assents on the implementation day one from the children whose parents consented. Since children were unable to comprehend written information, their oral assent was collected. video recorded.

KIBO

The DevTech Research Group (http://sites.tufts.edu/devtech/) at Tufts University developed KIBO for children from four to six years of age. KIBO’s block programming language is composed of 21 individual wooden programming blocks. Some of these blocks represent simple motions for the robot such as move forward, backward, spin, and shake (Sullivan & Bers, 2016). Other blocks represent complex programming concepts such as Repeat Loops and Conditional “If” statements that involve sensor input (Sullivan & Bers, 2016). KIBO allows children to program moves using interlocking wooden programming blocks. KIBO wooden blocks have embedded scanners that allow users to scan barcodes on the programming blocks and send a program to the robot instantaneously (Fig. 1).

Implementation

A robotics station was implemented in a prekindergarten classroom for a 2-week period, with the station being treated the same as other stations in the classroom. A total of seven stations were set up for free play time, which lasted approximately 40 min per day, including the robotics station, which

Fig. 1 KIBO set

was one of the required stations along with reading and puzzle/math stations. Children were allowed to rotate among the stations based on their interests. Activities during the first week focused on motivating children's interest, introducing basic programming information, and teaching how to operate the KIBO. During week 1, researchers introduced KIBO in day 2 and placed KIBO in robotics station from day 2 till the end of project implementation. At least one researcher stayed in the robotics station for 2 weeks to provide assistance as needed. In the second week, children played with KIBOs in the robotics station during free play time. Details about the activities during each week are provided in Table 1.

In Week 2, three sets of KIBO were placed in the robotics station. The KIBO allowed children to build and program using programming wooden blocks. When implementing KIBO, action blocks (e.g., Forward, Backward, Turn Right, Turn Left, etc.) were included for the first two days and sense blocks (e.g., hearing, seeing blocks) and one condition block ("Wait for Clap") were included from day 3 of Week 2. The condition block contains a sensor that senses when a child claps. After sensing clapping, KIBO follows the next command.

Measures

In order to evaluate the children's programming skills, individual interviews were conducted using KIBO. To avoid any directional confusion, the researcher and child sat side-by-side during the interview. These interviews were one-on-one with a researcher and lasted for 10–15 min each. All interviews were video recorded. Two researchers of this study reviewed video recording together and transcribed it. After transcribing, they analyzed the transcript to check the accuracy of children's programming tasks. Table 2 shows the interview questions used, which included six problem tasks involving copying programming (1 task), programming (2 tasks), programming with conditional statements (2 tasks), and decoding (1 task).

Results

Table 3 shows children's completion rates on the four tasks: copying programming, programming, programming with a condition statement, and decoding.

Copying Programming

As Table 2 shows, all but one child was able to successfully complete the copying programming task (Begin–Forward–Backward–Shake–End). The child didn't use "Begin" and "End" blocks, which caused a non-movement of the KIBO.

Table 1 Weekly activity information

Week 1
Day 1: Introduced children to "Nao," a humanoid robot, and presented Nao's several preset movements with a songs or music during whole group activity. Children followed Nao's movements
Day 1: Invited a guest speaker from a university robotics lab to the class who brought "Pepper," a humanoid robot that can interact with children with basic conversations (e.g., what's your name? how old are you? what's your favorite color? etc.).Read a children's book (*How to Code Sandcastle* by Josh Funk)
Day 2: Introduced KIBO with wooden action blocks to practice scanning
Days 2–5: Set up a robotics station-programming activity with a hand-made puzzle[a] and one set of KIBO for children to explore
Week 2
Days 1–2: KIBO with action blocks
Days 3–5: Adding sense and condition blocks

[a]Each puzzle is composed of 2 dice (a number die, and a direction die) and a game board which is a grid with squares. The game board includes a "Begin" square and "robot shelter" square (see "Appendix 1")

Table 2 Programming interview tasks

Programming interview tasks	Interview tasks and questions
Copying given programming	Show the process of programming the KIBO, ask the child to copy the exact programming. See below for the example (Begin–Forward–Backward–Shake–End) 
Programming (2 sets of programming tasks)	Question1. Ask the child to program the KIBO using 4 programming blocks and ask the child to explain what they were to program and have them program Question 2. Ask the child to program another way and ask them what they were to program and have them program Note. Successful completion means that the programmed outcome matches the child's plan
Programming with condition statement	Ask the child to program the KIBO using four blocks and one additional block ("Wait for Clap" conditional programming block). If a child claps, the robot continues to follow the program
Decoding (interpreting programming)	Ask the child to show the programming blocks by seeing how KIBO moves. See below for the sequence of programming (Begin–Forward–Beep–End) 

Programming

Thirteen children successfully completed two sets of programming tasks based on their plans. One child was only able to successfully complete one program task. The child programmed one task accurately but the other one inaccurately, so the programmed KIBO didn't match the program the child had planned (Begin–Forward–Spin–End) and the programmed outcome (Begin–Forward–Backward–End). The child used "Backward" block instead of "Spin" blocks.

Program with a Condition ("Wait for Clap") Statement

Seven children successfully completed the program with condition tasks using the "Wait for Clap" block and performed "clapping" for the next command to work. Seven children did not successfully complete the tasks (see below for the Table 4) because they were missing either the condition block (n = 2) or the condition command (n = 5, missing "clapping").

As Table 3 presents, most children accurately programmed and copied programming. However, there were common errors made on a condition statement by placing or commanding a condition action (clapping) to move KIBO for the next command. Condition block (clapping block) requires the children to clap to move forward to the next action. When two children did not place an action block (clapping), KIBO moved based on action blocks. However, when five children missed clapping though they placed a condition block (clapping block), KIBO stopped movements (See Table 4).

As reported in Table 4, three children's programming didn't work resulting in non-movement of KIBO. After researcher's facilitations (researcher pointing out the condition block-clapping, two children were able to add condition action (clapping) to have KIBO move for the next steps. One child was looking at the researcher when the researcher pointed out the condition block and didn't perform condition action.

Table 3 Task completion rates

Tasks	Number of children who completed the task/total number of children	Number of children who didn't complete the task (or incorrectly complete the task)/total number of children
Copying programming	13/14 (92.86%)	1/14 (7.14%)
Programming	13/14 (92.86%)	1/14 (7.14%)
Programming with a condition statement	7/14 (50.00%)	7/14 (50.00%)
Decoding	10/14 (71.44%)	4/14 (28.56%)

Table 4 Programs with conditional statements

Program errors	Programming sequence (KIBO movements)
Missing condition blocks Two children didn't use the condition block and KIBO moved as programmed	Child 1 Begin–Forward–Forward–Forward–End Child 2 Begin–Forward–Backward–Spin–End
Missing condition command Five children included a condition block but didn't clap to command the KIBO for the next movement	Child 3 Begin–Wait for Clap–Forward–Backward–Spin–End (KIBO didn't move) Child 4 Begin–Wait for Clap–Forward–Backward–Spin–End (KIBO didn't move) Child 5 Begin–Forward–Backward–Wait for Clap–Spin–End (KIBO moved forward and Backward) Child 6 Begin–Forward–Wait for Clap–Backward–Backward–End (KIBO moved Forward) Child 7 Begin–Wait for Clap–Forward–Backward–Spin–End (KIBO didn't move)

Table 5 Decoding

Task Correct decoding order: Begin–Forward–Beep–End	Number of children
Begin–Forward–Beep–End	10 (71.44%)
Forward–Beep–End	1 (7.14%)
Begin–Forward–Beep	1 (7.14%)
Forward–Beep	2 (14.28%)

Decoding

As presented in Table 5, ten children accurately decoded a task by identifying the programming sequence (Begin–Forward–Beep–End). Four children missed one or more commands such as “Begin” ($n=1$), “End” ($n=1$) or both ($n=2$).

As presented in Table 5, the common error(s) children made is placing “Begin” or/and “End” block. Since these two blocks don't actually involve specific actions besides “Begin” or “End,” children often make a mistake omitting these two blocks.

Conclusion and Discussions

The study found a significant improvement in all areas of programming skills, including copying programming, programming, programming with a conditional statement, and decoding, after playing with the robot (KIBO). Robotics presents an innovative approach for early childhood educators to introduce young children to programming/coding concepts that are frequently overlooked in early childhood STEM education. It provides children with hands-on experiences to practice and acquire fundamental programming skills (Di Lieto et al., 2016). According to the results of this study, more than 90% of the participating children were able to complete two programming tasks utilizing a codable robot (KIBO) by accurately copying a program and executing two different types of programming. However, one child was unable to complete both tasks, including the copying programming task and one of the programming tasks. This child has a specific language impairment (SLI) and receives language therapy on a weekly basis. Since SLI may affect a child's communication abilities, including speaking, listening, reading, and writing, programming KIBO that involves visual representations of communication skills could have a negative impact on the child's performance in copying and programming tasks. It is recommended that further research be conducted to explore whether and how the child's SLI affects their programming skills.

This study highlights the importance of providing young children with opportunities to learn and practice programming skills through robotics. The findings suggest that even young children can successfully complete programming tasks using a robot like KIBO. This is in line with a previous finding, indicating that children as early as 3 and 4 years are able to master basic programming skills (Bers, 2010; Sullivan & Bers, 2016). As children grow, they are able to master more complex programming skills (Sullivan & Bers, 2016). Additionally, the study highlights the importance of debugging as a critical skill to emphasize in computer science education, as it enables children to identify and correct errors in their programming. It is also important to note that debugging can help children manage their emotions, such as frustration, when faced with challenges. Teachers can support both cognitive growth and socio-emotional development by providing opportunities for debugging and problem-solving in their programming activities. Overall, this study highlights the potential of robotics in early childhood education to promote programming and problem-solving skills.

The study has uncovered another promising finding, highlighting the importance of conditional statements in programming. A conditional statement allows a robot to execute different actions based on the condition. This is closely related to "if" statements, where action I takes place if a command is true, and action II if it is false (Project Lead the Way [PLTW], n.d.). In the study, only 50% of children were able to program the KIBO robot with the conditional statement "Wait for Clap," which would make the KIBO follow the next programmed command after hearing a clap. During interviews, some children missed the conditional block or didn't clap to trigger the follow-up action. Although it is a complicated concept, it is a crucial element of coding and robotics. Engaging children in daily unplugged activities during early childhood can help promote their understanding of conditional statements (Lee & Junoh, 2019). For instance, a teacher could say, "I'll choose a child to be my helper today who is sitting quietly on their chair and looking at me." This involves a conditional statement: if a child is quiet and attentive, they might be selected by the teacher. Another popular activity that employs a conditional statement is "Simon Says", in which children perform an action when they hear "Simon".

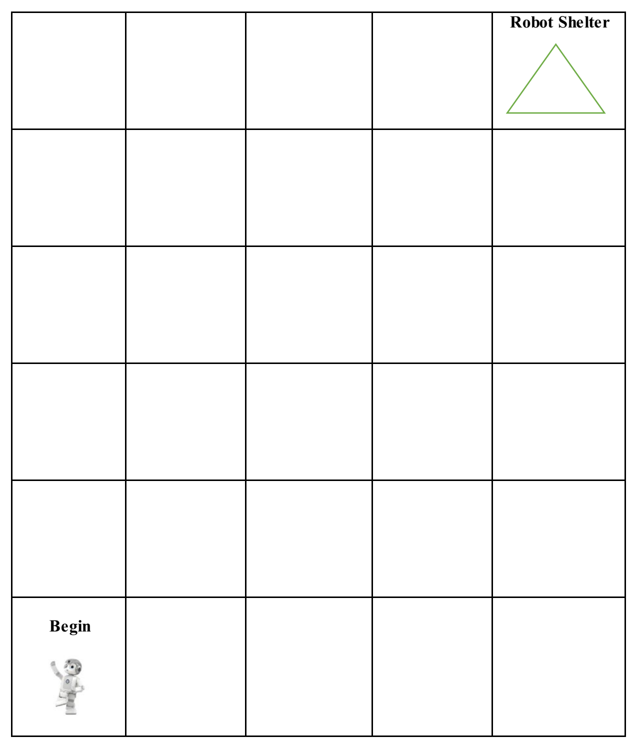
The study demonstrates that even children as young as three or 4 years of age can acquire programming skills by engaging with basic programming concepts. Integrating age-appropriate, tangible robots into the learning process provides children with opportunities to master fundamental and concrete programming skills.

Study Limitations

While the results of this study show promising findings regarding children's programming capability after engaging with robots as a station activity, there are certain limitations that should be acknowledged. First, it is important to note that this study is exploratory in nature and involved a limited sample of only fourteen children. The evaluation of the impact of robot interaction in the station was based on pre-and post-assessments. However, in this current paper, we did not include children's discourse. The absence of children's discourse during the station activity hinders the acquisition of rich data on their reasoning and thinking processes. Nevertheless, we plan to analyze the data for a follow-up study, where we will delve into the children's discourse to gain a deeper understanding of their interactions and learning experiences. Future research should aim to investigate children's interactions with robots in their daily lives to gain a deeper understanding of their thinking processes. Second, the small sample size restricts the generalizability of the findings to the wider population. To enhance the validity of the results, it is essential to conduct a quantitative study with a larger number of children.

Appendix 1

Robotics Station for Programming Activity with a Puzzle



References

Alimisis, D. (2012). *Robotics in education & education in robotics: Shifting focus from technology to pedagogy.* Retrieved from https://roboesl.eu/wp-content/uploads/2017/08/Robotics-in-Education-Education-in-Robotics.pdf

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice, 12*(2), 10–22.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Oxford University Press.

Bers, M. U. (2018a). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge Press.

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

Cherniak, S., Lee, K., Sung, E., & Jung, S. E. (2019). Child-identified problems and their robotics solutions. *Journal of Early Childhood Research, 17*(4), 347–360.

Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education, 7*(39), 1–18. https://doi.org/10.1186/s40594-020-00238-z

Di Lieto, M. C., Pecini, C., Castro, E., Inguaggito, E., Cecchi, F., Dario, P., Cioni, G., & Eguchi, A. (2016). *Computational thinking with educational robotics*. In G. Chamblee, & L. Langub (Eds.), *Proceedings of society for information technology & teacher education international conference* (pp. 79–84). Association for the Advancement of Computing in Education (AACE). Retrieved from https://www.learntechlib.org/p/172306.

Eguchi, A. (2014). Educational Robotics for Promoting 21st Century Skills. *Journal of Automation Mobile Robotics and Intelligent Systems, 8*, 5–11. https://doi.org/10.14313/JAMRIS_1-2014/1

Flannery, L. P., & Bers, M. U. (2013). Let's dance the '"robot hokey-pokey!"': Children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education, 46*(1), 81–101.

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., & Carboni, A. (2022). Educational robotics intervention to foster computational thinking in preschoolers: Effects of children's task engagement. *Frontier in Psychology, 12*. https://doi.org/10.3389/fpsyg.2022.904761

Govind, M., & Bers, M. (2021). Assessing robotics skills in early childhood: Development and testing of a tool for evaluating children's projects. *Journal of Research in STEM Education, 7*(1), 47–68.

International Society for Technology Education and the Computer Science Teachers Association. (2011). *Operational definition of computational thinking for K-12 education.* Retrieved from https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Kazakoff, E., 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. https://doi.org/10.1007/s10643-012-0554-5

Keren, G., & Fridin, M. (2014). Kindergarten Social Assistive Robot (KindSAR) for children's geometric thinking and metacognitive development in preschool education: A pilot study. *Computers in Human Behavior, 35*, 400–412. https://doi.org/10.1016/j.chb.2014.03.009

Lee, J. (2020). Coding in early childhood education. *Contemporary Issues in Early Childhood, 21*(3), 266–269. https://doi.org/10.1177/1463949119846541

Lee, J., Joswick, C., & Pole, K. (2023). Classroom play and activities to support computational thinking development in early childhood. *Early Childhood Education Journal, 51*(3), 457–468. https://doi.org/10.1007/s10643-022-01319-0

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal, 47*(6), 709–716. https://doi.org/10.1007/s10643-019-00967-z

Lott, K., Urbanek-Carney, S., & Mitchel, A. (2019). Engineering encounters: Cookie jar alarms early childhood engineering with robotics. *Science and Children, 57*(3), 66–72.

National Association for the Education of Young Children. (n.d.). *Effective strategies to support child-centered learning.* Retrieved from https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/pubs/transforming-teaching-chap-2-excerpt.pdf

Piaget, J. (1953). *The origins of intelligence in the child*. Routledge.

Project Lead the Way. (n.d.). *Robotics reference guide*. Retrieved from https://www.keysschools.com/cms/lib/FL02202360/Centricity/Domain/2883/The%20Entire%20Robotics%20Reference%20Guide.pdf

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education, 169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education, 14*, 1–15. https://doi.org/10.1186/s41239-017-0080-z

Strawhacker, A., & Bers, M. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development, 67*(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education: Innovations in Practice, 15*, 145–165.

Sullivan, A., & Bers, M. U. (2019). Computer science education in early childhood: The case of Scratch Jr. *Journal of Information Technology Education Innovations in Practice, 18*, 113–138. https://doi.org/10.28945/4437

Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer programming with early elementary students with Down Syndrome. *Journal of Special Education*. https://doi.org/10.1177/0162643417704439

Texas Education Agency. (2021). *Prekindergarten registration information (20–21)*. Retrieved from https://tea.texas.gov/academics/early-childhood-education/prekindergarten-registration-information-21-22

Vizner, M., & Strawhacker, A. (2016). *Curious construction kit: A programmable building kit for early childhood*: *Proceedings of the 6th annual conference on creativity and fabrication in education (FabLearn)*, Stanford, CA, October 14–16. ACM.

Wing, J. (2011, Spring). Research notebook: Computational thinking—What and why? *The Link Magazine*, Carnegie Mellon University. Retrieved from https://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.