How Programming Helps Early-Age Learners Tackle Complex Concepts

Josipa Lipovac, Dorotea Potoč, Tomislav Jagušt
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb
{josipa.lipovac, dorotea.potoc, tomislav.jagust}@fer.hr

Abstract. *In this paper, we introduce a novel approach to facilitate the acquisition of complex learning concepts among early-age learners, which are typically designed for older age groups, through programming workshops. Our research demonstrates how 3rd-grade elementary school students can effectively comprehend and apply material typically taught in the 6th-grade curriculum, specifically related to the coordinate system and its navigation. We conducted a series of three iteration tests, with the first two being identical tests administered three months apart. The third test was intentionally designed to be more challenging and intricate to evaluate the student's ability to handle more difficult tasks after an additional three months. Our findings indicate that younger students can successfully grasp the fundamental concepts of the coordinate system and its navigation by utilizing Scratch*[1] *programming and engaging in enjoyable exercises, such as homework assignments.*

Keywords. complex learning concepts, programming, scratch, computational thinking

1 Introduction

In today's rapidly evolving digital world, programming has become one of the most important skills of the 21st century. As a result, more and more schools are introducing programming into their curriculum, and research shows that even young children can develop computational thinking skills through early exposure to programming languages and screens (Papadakis, 2022). Programming can be introduced as a standalone subject, or its foundations can be integrated into fun preschool games, music, dance, and other activities, which can help children develop the thinking skills necessary for success in the digital age (Macrides et al., 2022).

Learning programming can help children stay up-to-date with the latest technology trends, ensuring that they are not left behind in the ever-evolving tech landscape. In addition to learning how to program, children can develop other educational outcomes such as mathematical problem-solving, critical thinking, social skills, self-management, and academic skills (Popat and Starkey, 2019). By learning to code, children can quickly advance and understand and solve problems they may have never encountered before. Although the initial results may be poor, after the learning process, the material becomes clear, and they are able to solve tasks with increased accuracy (Pérez-Marín et al., 2020).

There are several reasons why children are eager to learn coding. First and foremost, it provides them with a sense of empowerment and a feeling of control over the technology that they use on a daily basis. Sometimes children become demotivated by failure if it is more difficult for them to master a programming task or to correct a programming error. Therefore, it is necessary to encourage children in fun ways to adopt a way of thinking, and not to be afraid of trial and error, because this is the main way of learning. Additionally, coding is a creative and dynamic process that encourages children to think critically, problem-solve, and experiment with different solutions. By mastering programming, children can achieve a wide range of outcomes, such as enhancing their mathematical and computational thinking skills, developing their social and emotional intelligence, and gaining a better understanding of how technology works and how it can be used to solve real-world problems (Ryan, 2021). With the ever-increasing importance of technology in modern society, the ability to code has become a highly sought-after skill that can open up a world of possibilities for children and set them on a path toward future success.

This paper shows how programming can help early-age learners understand complex learning concepts that are typically intended for older age groups, through programming workshops in Scratch. Specifically, our study focuses on 3rd and 4th-grade elementary school students and how they can effectively comprehend and apply material related to the coordinate system and navigation that is usually taught to 6th-grade students.

[1] https://scratch.mit.edu/

All of the students that has been on the workshops are 9 or 10 years old, and are fluent in reading.

The paper discusses the benefits of teaching computational thinking to young children and how it can help develop mathematical skills. We first review related work in this area in the section Related Work, then describe the workshops we conducted, and the terminology the children have learned. In the section Results, we present the exam results from three iteration tests: before any topics were introduced, after three months of learning, and at the end of the workshops more challenging tests to evaluate their understanding. The first two tests were identical, administered three months apart, and the third was more challenging and intricate. We discuss and interpret the results before concluding with a summary of our findings.

2 Related work

Computational thinking is considered a desirable competency for adapting to the future, and it is essential to introduce it to children at an early age so that they can adapt and grow alongside the technology. Schools use a variety of methods to promote computational thinking, such as introducing activities such as robotics, program design, and game-based learning. Children learn to cooperate and think differently, connecting concepts more easily. The most widely accepted concepts are project-based learning, problem-based learning, cooperative learning, and game-based learning. Less common but still valuable approaches include storytelling, design-based learning, and aesthetic experience (Hsu et al., 2018).

Research studies have shown that young children can benefit from learning programming and computational thinking skills in various ways. For example, using tablets or robots in storytelling activities can promote the development of skills needed in the 21st century, such as creativity, problem-solving, and digital competence, as well as enhance computational thinking skills (Yang et al., 2023). ScratchJr is also a tool that can promote computational thinking skills among young children (Stamatios, 2022).

Moreover, some studies have demonstrated that young children can effectively solve complex problems by breaking them down into smaller tasks. Using the Bee-Bot robot, children can develop computational thinking skills (Angeli and Valanides, 2020). It has also been found that children between the ages of 2 and 4 can learn thinking skills with Bee-Bot, perhaps with the help of parents or educators (Critten et al., 2022).

Introducing programming in early childhood education using robotics kits can foster better communication, collaboration, and creativity, and promote confidence in children to try new things without fear (Bers et al., 2019). Research also suggests that children as young as 8 years old can learn programming skills, and there are ongoing studies on whether programming should be introduced at an earlier age and when it should be taught (Duncan et al., 2014).

Scratch, as a programming tool, can help young children develop problem-solving skills through game design activities. Children can identify problems more easily, think more creatively, and engage in problem-solving attempts (Erol and Çirak, 2022).

Furthermore, the introduction of programming and robotics in schools can promote STEM education, and introducing programming into elementary education can foster computational thinking skills (Alam, 2022). Humanoid robots can also help children learn computational thinking naturally if included in the curriculum (Chen et al., 2017).

Additionally, studies have shown that using Scratch to teach programming can enhance computational thinking and improve mathematical understanding. Students' perception of computer science as a discipline changes, and they can switch to more challenging programming languages because they have already developed thinking and mathematical skills (Grover et al., 2015).

Through workshops, simple and fun activities, games, and the use of various tools such as LEGO robotic sets, Kodu environments, and mechanical calculators, children as young as 8 years old can learn abstract concepts typically taught to older children or even 18-year-olds. These activities provide a hands-on and interactive approach to learning, enabling children to explore and experiment with concepts in a playful and engaging way. By incorporating tools and technologies, children are exposed to real-world applications of abstract concepts and can develop a deeper understanding of their relevance and practical use. This approach to education has the potential to foster a lifelong love of learning and a curiosity for the world around them (Sović et al., 2014).

3 Workshop description

3.1 Scratch environment

Scratch is an innovative programming language and environment developed by the Lifelong Kindergarten Group at the MIT Media Lab. It is designed as a block-based system that empowers users to create interactive, media-rich projects (Maloney et al., 2010). The visual programming interface of Scratch allows beginners to concentrate on programming logic without being overwhelmed by syntax errors. By utilizing a drag-and-drop approach with vibrant and colorful coding blocks, Scratch becomes accessible and user-friendly, particularly for young children who can easily comprehend and manipulate it.

At its core, Scratch aims to provide a platform that encourages children to actively create interactive stories, games, and animations. It follows a structured learning approach to introduce programming concepts gradually. Starting with simple commands, users can gradually progress to more advanced ideas, building a solid foundation in programming.

Moreover, Scratch extends beyond programming and serves as a tool for the intuitive exploration of various mathematical concepts. It fosters a deeper understanding of mathematical principles, allowing children to engage with mathematics in a creative and interactive manner.

Furthermore, Scratch boasts an online community comprising both students and educators. This community serves as a supportive environment where children can share their projects, receive feedback, and collaborate with others.

3.2 Workshops organization

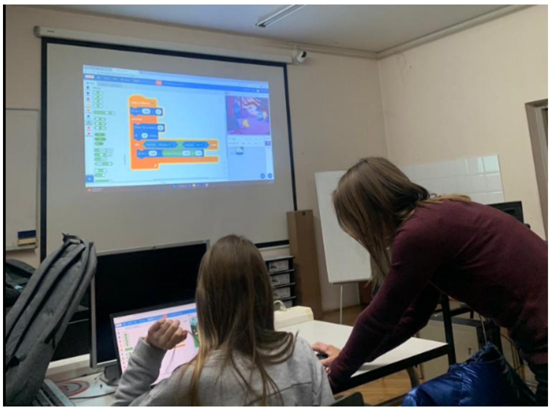

Figure 1: Image from workshop

The workshop program aims to familiarize young children in the early years of elementary school with the fundamental principles of programming through the Scratch environment. Students from the 3rd and 4th grades participated in a series of workshops spanning 22 weeks.

Table 1: Weekly workshops plans

WEEK	DESCRIPTION
1.	Getting to know the Scratch environment
2.	Coordinate system
3.	Directions
4.	Directions + Motion
5.,6.	Motion, Looks, repeat, forever and if-then
7.,8.	Variables, Operators and if-then-else
9.	Quiz
10.	Advanced motions and operators
11.,12.,**13.**	Project: merging learned knowledge
14.,15.	Clones
16.,17.,18.	Advanced programming concepts practice
19.,20.	Messages
21.,**22.**	Advanced programming concepts practice

Table 1 provides a breakdown of the concepts covered each week. Each workshop session lasted 45 minutes. Following certain workshops, students were assigned additional homework, which would then be thoroughly discussed in the subsequent workshop. On the link[2], it is possible to find projects that were worked on in certain weeks. Bolded in the table are the weeks during which survey tests were conducted, assessing the children's advancements in comprehending complex mathematical concepts.

Over the course of 22 weeks, the children were introduced to the Scratch environment, fundamental programming concepts, and various mathematical concepts that children usually encounter at school at an older age. As early as the second week, they were taught about the number line, negative numbers, and the coordinate system. They further expanded their understanding through assigned homework. In the third week, they explored the concept of a complete circle, learning the measurement of degrees and how they determine direction.

This knowledge of coordinates and directions was then applied in the creation of interactive games, providing a practical understanding of basic programming concepts. The workshop focused heavily on comprehending conditions, loops, and variables, while also delving into advanced Scratch programming topics like clones and messages.

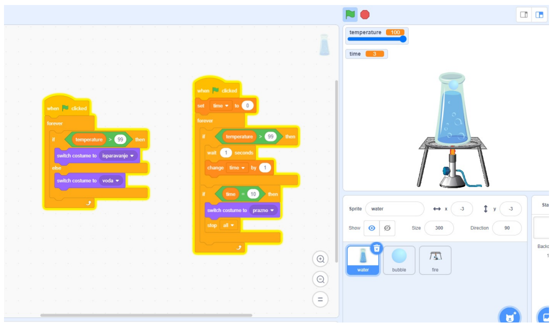

Figure 2: Example of the task on the workshop

4 The research results

4.1 The first test

At the start of the second workshop, prior to any lectures, children were administered the initial test. Its purpose was to evaluate the children's prior knowledge. The initial test consisted of three informative questions designed to check the children's familiarity with Scratch or programming. Throughout the entire semester, a total of 10 children participated in the workshops, but the comprehensive test results are available for only 6 of them.

The questions were:

1. Have you ever programmed in Scratch before these workshops?

[2] https://scratch.mit.edu/studios/33361306

2. Have you ever participated in STEMALICA's workshops before?

3. What is your knowledge of mathematics in school? What grade did you receive at the end of last school year?

4. Draw a number line in the space below this question.

5. If you have managed to draw the number line below question 4, mark the numbers 1.5 and 3.5 on it.

6. Write one negative number.

7. Determine the coordinates of points A and B marked on the coordinate system.

Based on the test results, five participants had prior programming experience with Scratch, and four had attended previous workshops by STEMALICA. Out of the group, five had an A grade in math, while one had a B. Only one participant successfully drew the correct number line, another attempted it, and the remaining participants left it blank. None of them managed to solve the fifth task, which involved marking decimal numbers. The same participant who drew the correct number line also correctly wrote a negative number. One participant wrote a positive decimal number, another wrote a positive integer, and the rest left it blank. None of the participants even attempted to solve the final task.

4.2 The second test

After 11 workshops, we conducted a follow-up test similar to the previous one, but without informative questions. This time, five participants successfully drew a correct number line, but only three of them accurately marked decimal numbers on it. All participants correctly wrote one negative number. Four of them successfully completed the final task. One participant correctly determined one point, but struggled with the other, while another participant didn't manage to complete the task accurately.

4.3 The third test

The examination took place on May 4th, 2023. The test consisted of seven tasks covering material on the coordinate system, number line, and negative numbers. An example task from the exam is shown in the **Figure 3**, where children had to determine the coordinates of treasure on the coordinate system.

The test questions were:

1. Please round off the negative numbers from the given list of numbers: 1.5, -9, 7, 2.25, -1, 0.5, -1.5, 2, 6.2, -5.3.

2. Please mark the following points on the coordinate system: A(1,1), B(6,2), C(3,6), D(4.5,2), E(1,3.5).

3. We are in the coordinate system, draw it. The point (-2, 4) is given, move 4 units to the right, and then 2 units down. What are the coordinates of your new point?

4. Please draw a coordinate system. The point (1,2) is given, move 3 units in the direction of -90°. What are your new coordinates?

5. Points A(2,2), B(2,5), and C(5.5,2) are marked on the coordinate system. To complete the rectangle ABCD, we need the coordinates of point D. What are its coordinates?

6. Determine the coordinates of points A and B marked on the coordinate system.

7. What are the coordinates of the treasure?

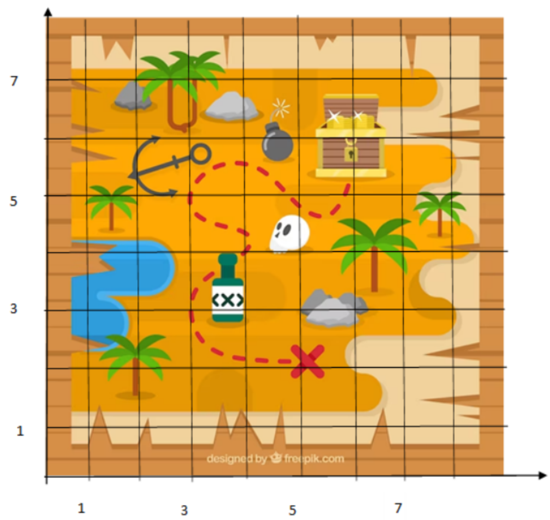

Figure 3: Example of the task on the test

All participants successfully completed the 1st, 6th, and 7th questions. The majority of participants faced difficulties with the 2nd task, but five of them managed to correctly mark at least one point, with two participants making only one mistake involving decimal numbers. Only one participant successfully completed the 3rd question, while three participants mistakenly interchanged the x and y coordinates, leading to incorrect end point. Similar issues were observed in the 4th task, with three participants accurately determining the direction despite the confusion in coordinate swapping. In the 5th task, four participants correctly marked the point, but only one of them successfully determined the point's coordinates, while others continued to struggle with decimal numbers. One participant misunderstood the task, and another participant correctly determined the decimal coordinate but not the integer one.

Table 2: Percentage of test results and participant's attendance

	1st results [%]	2nd attendance [%]	2nd results [%]	3rd attendance [%]	3rd results [%]
P1	0	85,62	100	86,36	36,67
P2	0	92,31	50	95,45	66,67
P3	50	100	75	100	73,33
P4	0	100	100	90,91	73,33
P5	0	100	62,50	68,18	46,67
P6	0	92,31	62,50	90,91	33,33

4.4 Attendance and test results

Table 2 presents individual participants' results for each test and their attendance records. Our goal was to examine the relationship between these two features.

Table 3: Average scores by tasks

	Test 1 [%]	Test 2 [%]	Test 3 [%]
1.	-	-	100
2.	-	-	43,33
3.	-	-	25
4.	16,67	83,33	25
5.	0	50	54,17
6.	16,67	100	100
7.	0	83,33	100

Table 3 displays the average test scores categorized by tasks. It is important to note that the 7th task in tests 1 and 2 is identical to the 6th task and very similar to the 7th task in test 3.

5 Discussion

Looking at results from Table 2 and evaluating the understanding of materials designed for older children it can be concluded that the participants exhibited varying levels of comprehension and skills. The analysis of their results highlights the importance of an individualized approach and adaptation of instructional materials to promote optimal learning.

Overall, certain participants demonstrated significant progress and exhibited a deep understanding of the concepts. These individuals successfully completed tasks, applied correct concepts, and accurately utilized mathematical notations. Their results indicate abilities in critical thinking, logical reasoning, and the application of learned knowledge.

On the other hand, some participants encountered challenges and made various errors. These individuals lacked an understanding of the materials and struggled to apply the concepts appropriately. Mistakes included coordinate substitutions, incorrect point labeling, and difficulties with decimal numbers. Additionally, a lack of concentration and impulsiveness were observed, resulting in imprecise responses.

These findings highlight the need for adapting instructional materials to provide participants with practical challenges tailored to their understanding. An individualized approach and careful guidance could enhance material comprehension among participants facing difficulties. It is also important to encourage participation and active engagement to improve concentration and reduce impulsiveness. This emphasizes that the understanding and acquisition of mathematical concepts among children can vary, and the teaching approach is a significant factor in achieving success. Adapting instructional materials, employing an individualized approach, and promoting active participation can be crucial and effective methods for enhancing mathematics understanding among children.

Furthermore, the student who attended all the lectures performed very well on all the tests, indicating a strong understanding. The errors made, such as swapping x and y coordinates or miscounting cells, may suggest the earlier mentioned tendency toward impulsiveness.

Also, participants P1 and P6 who performed poorly on the third test demonstrated a partial understanding of the given requirements. They consistently interchanged the order of x and y coordinates and mislabeled the initial coordinate in the third and fourth tasks. However, they displayed adequate displacement and a close approximation to the correct solution. The confusion was caused by the picture of the coordinate system in the second and fifth tasks, which led to incorrect labeling and rounding errors. Despite some areas of confusion, the students demonstrated an understanding of rotation and the coordinate system. We would attribute these errors to a lack of concentration and impulsiveness.

We would like to single out participant P5 who had a high number of absences in the workshops, which affected his test results. His performance was slightly poorer, indicating a lack of understanding of the basic concepts, which could also be noticed on the last test where his mistakes cannot be called clumsiness, but rather misunderstanding, because he did not understand some tasks that needed to be done, he also did not understand how to independently draw a coordinate system, so he left most of the test blank.

In the beginning, children had little to no knowledge about the subject matter, and on the first test, they encountered questions such as: "What is a line?" or "What does a negative number mean?" However, in the final results, we can see that the children made significant progress and learned the basic concepts of the coordinate system.

Table 3 illustrates the overall group performance on the test tasks. After 11 weeks, it is evident that participants showed significantly greater success in solving the repeated first test. It is important to note that test 3 posed a significantly higher level of difficulty compared to tests 1 and 2. Nonetheless, progress is clearly evident. The most notable advancement is observed in the final task of the 1st and 2nd tests, which corresponds to tasks 6 and 7 in the 3rd test. In the last test, participants achieved a perfect success rate in solving these two tasks. On the other hand, the lowest success rate was observed in tasks 3 and 4 of the third test. The primary challenge in these tasks appeared to be related to the reordering of coordinate axes. Additionally, the infrequency of drawing a coordinate system independently during the workshops may have contributed to this issue.

Children with higher attendance demonstrate better understanding, although some may have achieved lower overall scores. Their responses indicate comprehension of the required tasks and approximate accuracy in their solutions.

6 Conclusion

This paper presents how Scratch programming workshops can help younger children grasp complex mathematical concepts and basic programming fundamentals typically intended for older children. By introducing children to fun block-based programming in Scratch, such as using games or everyday life scenarios like locating objects in a room using a coordinate system, children are able to overcome abstract and unfamiliar terms like number lines, coordinate systems, and angles.

Furthermore, the study reveals the positive impact of workshop attendance on understanding. Although this may not be fully evident in the final results, participants who attended most or all of the workshops demonstrated a better understanding of the subject matter and the task requirements. On the other hand, the participant with the lowest workshop attendance showed difficulties in understanding and did not attempt to solve certain tasks.

The test results indicate that the children have learned the basic concepts, although more challenging tasks were not fully mastered. However, progress can be seen when they actually understand the task at hand and what is expected of them, but make minor mistakes, such as rotating incorrectly in the coordinate system or miscounting steps for movement. This highlights the importance of consistent workshop attendance for improving comprehension and successful task completion.

As for future work, it would be interesting to explore how to further improve the Scratch programming curriculum for young children, perhaps by designing more challenging exercises that build upon the concepts they have already learned. Additionally, it could be beneficial to investigate how Scratch programming can be used to supplement other areas of early childhood education, such as creative expression. Finally, it would be worth exploring the potential impact of introducing Scratch programming workshops to children from different socio-economic backgrounds or cultural contexts.

Acknowledgments

We would like to thank the association STEMALICA[3], for the held workshops and help with the research.

References

Alam, A. (2022). Educational robotics and computer programming in early childhood education: A conceptual framework for assessing elementary school students' computational thinking for designing powerful educational scenarios. In *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, pages 1–7.

Angeli, C. and 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.

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

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., and Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers Education*, 109:162–175.

Critten, V., Hagon, H., and Messer, D. (2022). Can pre-school children learn programming and coding through guided play activities? a case study in computational thinking. *Early Childhood Education Journal*, 50(3):969–981.

Duncan, C., Bell, T., and Tanimoto, S. (2014). Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, WiPSCE '14, page 60–69, New York, NY, USA. Association for Computing Machinery.

[3]https://stemalica.com/

Erol, O. and Çirak, N. S. (2022). The effect of a programming tool scratch on the problem-solving skills of middle school students. *Education and Information Technologies*, 27:4065–4086.

Grover, S., Pea, R., and Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2):199–237.

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. *Computers Education*, 126:296–310.

Macrides, E., Miliou, O., and Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*, 32:100–396.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4):1–15.

Papadakis, S. (2022). Apps to promote computational thinking and coding skills to young age children: A pedagogical challenge for the 21st century learners. *Educational Process: International Journal*, 11(1):7–13.

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., and Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105.

Popat, S. and Starkey, L. (2019). Learning to code or coding to learn? a systematic review. *Computers Education*, 128:365–376.

Ryan (2021). Coding for kids: Reasons kids should get started, and how they can find success. `https://www.idtech.com/blog/5-reasons-your-child-should-learn-to-code`. Accessed: May 17, 2023.

Sović, A., Jagušt, T., and Seršić, D. (2014). How to teach basic university-level programming concepts to first graders? pages 1–6.

Stamatios, P. (2022). Can preschoolers learn computational thinking and coding skills with scratchjr? a systematic literature review. *International Journal of Educational Reform*.

Yang, W., Ng, D. T. K., and Su, J. (2023). The impact of story-inspired programming on preschool children's computational thinking: A multi-group experiment. *Thinking Skills and Creativity*, 47:101–218.