Elinvo (Electronics, Informatics, and Vocational Education)
Vol 10, No. 1, 2025, pp. 54-62
ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Investigating Computational Thinking in K–12 Visual Programming Activities on Code.org: A Brennan-Resnick Framework Approach

Cucuk Wawan Budiyanto[1*], Kristof Fenyvesi[2], Yustika Indah Maharani[3] , Rosihan Ari Yuana[4], Putri Khoirin Nashiroh[5] , Rizka Latifah[6]

[1,3,4,6]Universitas Sebelas Maret, Jalan Ir. Sutami 36A, Kentingan, Jebres, Surakarta, Jawa Tengah 57126, Indonesia
[2]University of Jyväskylä, Seminaarinkatu 15, 40014 Jyväskylän yliopisto, Finland
[5]Universitas Negeri Semarang, Kampus Sekaran Gunungpati, Semarang, Jawa Tengah 50229, Indonesia
[1]cbudiyanto@staff.uns.ac.id*; [2] kristof.fenyvesi@jyu.fi; [3]imyustika@student.uns.ac.id; [4]rosihanari@staff.uns.ac.id; [5]putrikhoirin@mail.unnes.ac.id; [6]rizka.latifah62@student.uns.ac.id
* corresponding author

Article Info

Article history:
Received October 21, 2024
Revised April 2, 2025
Accepted June 24, 2025
Available Online July 26, 2025

Keywords:
Block-based programming; Brennan-Resnick Framework; Code.org; Computational Thinking; qualitative approach.

Abstract

Global interest in Computational Thinking and learning to code has increased. Teaching elementary school students to code and develop computational thinking is a crucial skill for the 21st century. Code.org or Scratch are increasingly used by researchers and educators to evaluate the best practices in digital environment. Such an understanding leads to the urgency to investigate how the block-based programming environment contributes to the development of Computational Thinking and how the development takes place for K-12 students. This research departed from the previous study on the use of Brennan and Resnick framework to evaluate the development of Computational Thinking in various literature on visual programming. Employed a qualitative design, the research investigates K-12 students' response to the activity. The portfolio, then, analyzed using the Brennan and Resnick Framework for Computational Thinking development. The results demonstrated that six of the seven computational concepts could be taught in 10 stages using the "Dance Party" task. Work on the animation project in the tenth stage of "Dance Party" has well-prepared the four computational practices projects. This has a great deal to do with the student's ability to question perspective, as young people do not perceive a disconnect between the surrounding technology and their ability to negotiate reality.

This is an open access article under the CC–BY-SA license.

Corresponding Author:
Email: cbudiyanto@staff.uns.ac.id

INTRODUCTION

Global interest in Computational Thinking and learning to code has increased. Teaching elementary school students to code and develop computational thinking is a crucial skill for the 21st century [1-3]. Students in grades K-12 must master computational thinking in order to "think on many abstractions" [4-6]. Moreover, Computational Thinking aligns with numerous 21st-century competencies, such as problem-solving, creativity, and critical thinking [7].

K–12 students can be introduced to programming through unplugged computing, educational robotics, and visual programming to foster Computational Thinking skills. Unplugged computing is a method of teaching Computational Thinking that eschews dual devices in favour of beneficial instruments and activities. The opportunity to learn more about interacting with programmed or connected objects is the second advantage of using educational robotics as a teaching tool. To enhance

Computational Thinking, use visual programming to learn code in new programming environments, such as Scratch and ScratchJr [1].

Digital learning environment, such as Code.org or Scratch are increasingly used by researchers and educators to evaluate the best practices in digital environment [8-10]. The learning platform was designed to teach children as young as three years old programming fundamentals through entertaining online games [11]. Blocks on Code.org enable novices to structure program without becoming frustrated while attempting to connect the proper programming syntax [12]. In addition, Hour of Code (Code.org), the most extensive educational campaign in history, is conducted using Scratch [13, 14]. Code.org promotes its coding program in schools worldwide, and the initiative has successfully provided services to more than 500 million students [15]. Despite sheer interest in adopting Code.org for education, most of the literature, however, reports mainly based on the study of previously published literature [16-18]. Moreover, the block-based programming language has long been considered appropriate for preschool / primary school education [19-21]. Even some of the activities may not be appropriate for the early learner and tend to discourage their interest in programming [22]. Such an understanding leads to the urgency to investigate how the block-based programming environment contributes to the development of Computational Thinking and how the development takes place for K-12 students.

This research departed from the previous study on the use of Brennan and Resnick framework to evaluate the development of Computational Thinking in various literature on visual programming [23], pertinent to conceptualizing interactive media design in a visual programming activity. Based on their research, activities like Scratch, Brennan, and Resnick have created a framework for the Computational Thinking evaluation technique for young people. The framework consists of three dimensions: computational perspectives, computational exercises, and computational ideas [9, 24].

Despite growing efforts to promote computational thinking (CT), little is known about how early programmers develop these skills or how best to assess them [22]. As noted by Macrides et al. [16], while programming is increasingly introduced in early childhood education, there remains a lack of clarity on effective assessment practices and pedagogical approaches tailored to young learners. Current research highlights the need for assessment tools that move beyond code accuracy to capture key CT processes like problem decomposition, algorithmic thinking, and pattern recognition [25]. This article addresses this gap by investigating, using the Brennan and Resnick framework, how Code.org influences K-12 students' development of Computational Thinking within a visual programming environment. We further explore the connection between these acquired skills and K-12 students' problem-solving abilities.

METHODS

This study employs a qualitative research methodology, specifically a case study design [26], to gain an in-depth understanding of how K–12 students develop computational thinking through a visual programming activity. A qualitative case study was deemed appropriate over experimental or mixed-methods approaches, as the focus of this research is not to measure outcomes or generalise findings but rather to explore students' learning experiences, behaviours, and meaning-making processes in a naturalistic educational setting [27]. Thus, the purpose of the case study aligns with the research objectives: to gain a deeper understanding of the relationship between Code.org, computational thinking, problem solving, and K-12 students' learning experiences. The research data consist of student project documents and interview transcripts, in line with the case study approach, which emphasizes the use of multiple data sources collected over time. The data collection and analysis followed a structured sequence: (1) participants were selected using purposive sampling based on their digital literacy and interest in animation; (2) students engaged in a Code.org "Dance Party" programming task over a defined period, consisting of ten stages; (3) data were collected through analysis of project artifacts and semi-structured interviews guided by instruments aligned with the Brennan and Resnick framework; and (4) the collected data were analyzed thematically following Braun and Clarke's six-phase approach, identifying recurring patterns related to computational concepts, practices, and perspectives.

To ensure a targeted sample of participants, we employed purposive sampling [28]. Our focus was on student in grades K-12 who demonstrated proficiency in reading, technology (computers or smartphones), and had a keen interest in animation. The samples used were 1 junior high school student, 3 senior high school students, and 1 vocational high school student.

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Table 1. Brennan & Resnick Computational Concept Assessment Instrument

No.	Intrument	Indicator
1.	Sequences	Identify a series of steps for a task
2.	Loops	Execute the same sequence several times
3.	Parallelism	Make things happen at the same time
4.	Events	One thing that causes another thing to happen
5.	Conditional	Make decisions based on conditions
6.	Operators	Mathematical and logical expressions
7.	Data	Storing, retrieving, and updating values

The data collection methods used were document analysis and interviews. The participant will be assigned a "Dance Party" animation creation project on Code.org. The document analysis used is a student workspace containing several indicators computational concept [24] as seen in Table 1. The document produced from the animation project assignment is used to explore the computational concept applied by the participants. The interview will conduct after project assignment. The interviews used the computational practices and perspectives instruments [24] as seen in Table 2 and Table 3.

Data analysis on qualitative research is often carried out in conjunction with data collection. Generally, the data collected is in the form of the experiences and perspectives of participants. Because qualitative research seeks to obtain a detailed description of the object of study [30], this study uses thematic analysis techniques to process the collected data.

Table 2. Brennan & Resnick Computing Practice Assessment Instrument

No.	Intrument	Indicator
1.	Being active and incremental	Develop the concept a little, then try it out and then develop it some more
2.	Testing and debugging	Making sure everything works and fixing errors
3.	Reusing and remixing	Create something by developing what others or yourself have done
4.	Abstracting and modularizing	Building something big by putting together a collection of smaller parts

Table 3. Brennan & Resnick Computational Perspective Assessment Instrument

No.	Intrument	Indicator
1.	Expressing	Recognizing that computing is a creative medium
2.	Connecting	Recognizing the power of creating with and for others
3.	Questioning	Feel empowered to ask questions about yourself and your surroundings

According to Braun and Clarke [29], thematic analysis is a method that can be used to identify, analyze, organize, describe, and report themes found in data sets. This technique aims to place a piece, that is, a pattern in data that is important or interesting and use this theme to answer research [30]. Thus, the analysis results obtained from this study are themes that can answer research questions. Braun and Clarke [in 30] provide a six-stage guide to conducting thematic analysis.

RESULT AND DISCUSSION

The analysis results, which are also presented as a descriptive paragraph above, are then reconfigured into discussion paragraphs that answer the research questions more explicitly. The following are insights deduced from the Brennan and Resnick frameworks.

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Code.org Influence

According to the analysis of the collected data, Code.org significantly impacts students' learning experiences in grades K-12 by training their computational thinking skills. The "Dance Party" project on Code.org contains nine stages of concept training and one set of assignments that can be used to train nearly all indicators on aspects of computational concepts and practices. In addition, interviews were conducted to delve deeper into the computational perspectives of the participants. Following is a more in-depth discussion of how Code.org impacts students' learning experience and computational thinking skills.

Towards the student learning experience

Based on the interview transcript presented above, it is possible to infer how Code.org influences students' learning experiences as they practise Computational Thinking skills. Table 4 summarises all participant responses regarding the impact of Code.org on their educational experiences.

Throughout the early stages of working on such an animation project, students refer to the previous exercises and video tutorials, as shown in Table 4. One of the students elaborated on his initial steps: “I began by viewing the tutorial, continued to follow the existing tutorial and continued to practise it” (Participant 5). While the findings of this study align with Zha, et al. [32], who observed that embedded tutorials and structured exercises enhance students’ enjoyment and engagement with programming tasks, they contrast with critiques such as those by Lambić et al. [22]. Lambić and colleagues reported that younger elementary students often experienced frustration and disengagement when using Code.org, particularly due to the platform's limited adaptability to individual learning needs. This discrepancy may stem from contextual differences; the present study involved older students with basic digital literacy and specific interest in animation, which could have positively influenced their engagement. Additionally, the structured nature of the “Dance Party” module may have provided enough support for this age group to experience success without becoming overwhelmed.

During the middle phase of the animation project, participants encountered various challenges. At this stage, computational practices—particularly testing and debugging, as well as reusing and remixing—played a critical role in supporting their progress. Participants responded to these difficulties by drawing from prior examples, modifying elements of their projects, and systematically experimenting with various options until they identified a satisfactory solution.

Table 4. Student learning experience answers

Question Topics	Answer Summary
Getting Started with the Project	Four out of five students started their projects by reviewing the exercises and video tutorials that had been given previously. While the others simply followed the instructions given.
Facing Difficulties	Four out of five participants overcame their difficulties by trying each option one by one until they found the one they thought was right. Others, on the other hand, simply asked someone else.
Modifying or Following the Example	Three out of five participants tend not to follow the example given. Modifications are made by changing the character or different movements. However, there are also those who follow the example and also do both.
Completing the Project	All participants felt happy after completing their animation project.
What to like and what not to like about “Dance Party”	Things that are liked such as determining the character, choosing the song, seeing the results of the movement, and also the easy introduction of the concept. While for things that are not liked such as the narrow workspace, the process of assembling the blocks, and also the introduction that is too long because it requires 10 stages.

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

The student's confession includes the statement, “I live first when I encounter the most significant problems”. Similarly, (Participant 4) made changes, “selecting a different persona if the result is positive and is not subsequently erased”. Adapt the setting to the character's requirements (Participant 3). A similar study found that through this trial-and-error method, most young people apply their newly acquired knowledge to the design of projects [33].

In the final phase, after completing his project, the student expresses his emotions through questions related to the expressing perspective indicators. Despite the difficulties of the process, all students expressed satisfaction with their accomplishments. Similar research conducted by Zha, et al. [32] revealed that four out of six students characterised their Computational Thinking learning experience as pleasant and enjoyable. Nonetheless, one of the students also disapproves of the "Dance Party" project's ten stages, stating, "the first part is typically disliked. There are ten levels identical to the tutorial. Despite being thoroughly described, I find the ten stages excessively lengthy." Participant No. 5.

Thus, Code.org influences students' learning experiences throughout their project development's beginning, middle, and end. In the early phases of the project, video tutorials and exercises are provided for the students' benefit. Implementing testing and debugging practices, as well as reusing and remixing, assists students in overcoming obstacles they encounter during the mid-work phase. Through an expressing perspective, students express their opinions regarding the learning process.

Toward computational thinking ability

Computational concepts: Based on the findings of the analysis of the "Dance Party" project document, we would argue that not all indicators related to computational concepts are recommended to taught to K-12 students. Sequences, Loops, Events, Parallelism, Conditionals, and Operators are aspects of computational vision that can be taught in the "Dance Party" project. In this regard, the most intuitive concepts for K-12 students are Sequences, Loops, Events, and Equality.

This is evidenced by all participants scoring three on each of the aforementioned concepts. The participant receives a score of 3 if he or she can create a series of blocks from the idea and then successfully execute the series. Figure 1 depicts an example of a sequence of Participant 2's computational concept application blocks.

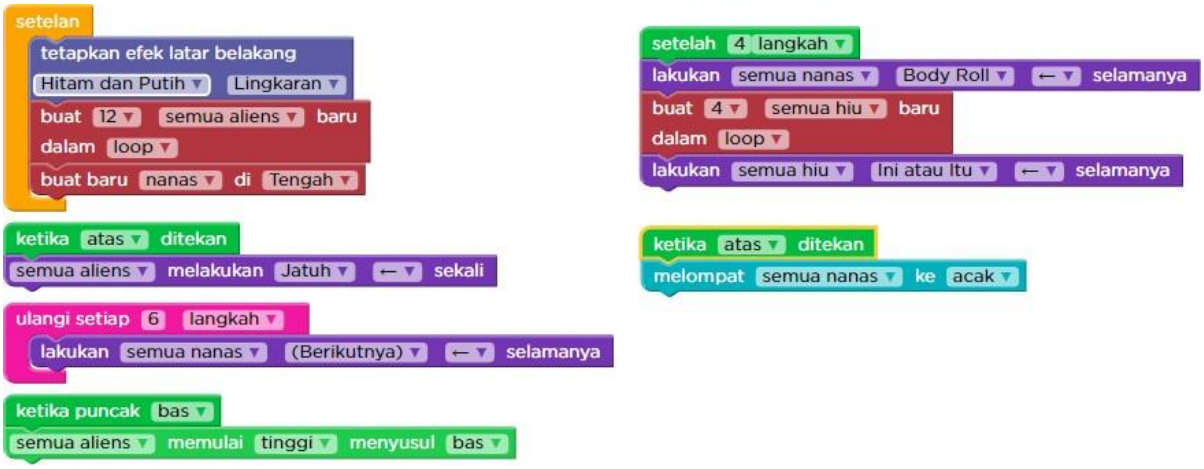

Figure 1. Participant block series 2

Computational Practices: Interviews with participants have yielded several insights regarding computational practices. The following are excerpts from the participants' responses regarding how they initiated their projects. "By reexamining previous instances" (Insertion 2). Beginning with viewing the tutorial, following the existing tutorial, and continuing to practise it" (Participant 5).

Based on the response, the practice of being incremental and iterative can begin by examining previous exercises. It is consistent with Litts, et al. [33]’s findings, which indicate that young people engage in being incremental and iterative as their project ideas develop, expand, and change before interacting with their projects.

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Testing and debugging techniques are taught by observing how participants overcome obstacles. Participant 3 said, “Try one out. If it is not suitable, try the other option”. Supported by Participant 4's response, “When I encounter the most obstacles, I walk first. It is desirable that it not recur if it is not deleted again”, he said.

The practice of testing and debugging can be trained by systematically attempting options until the most suitable one is identified. Similarly, the research conducted by Luo, et al. [34] revealed that the two participants of their study used a straightforward testing and debugging strategy involving code testing, changing what was necessary, and testing again.

The practice of reusing and remixing is honed through the decision to either replicate the example or make alterations. “Make alterations, create your own” (Participant 3). “No more so. I fiddle with myself. Keep the movements of these characters' components” (Participant 4). According to the excerpt from the previous response, participants modify their projects by changing something different from the example. The statement is also consistent with the findings of research conducted by Litts, et al. [33] who discovered that in some instances, developers reuse components from examples in their projects alone.

Participants are trained in abstracting and modularizing practices throughout their project work. Participant 1 said, “Lani penned the song first, then the suit, and then the character was the same on the back. Only the motion”. Similarly, participant 5 stated, “If I am first, it is similar to playing a song. Following the song, there is a bass-like buzz. Continue beating it. Here is where you can determine the movements and personality corresponding to the song.”

This goes against the natural method of practising abstraction and modularization that Brennan and Resnick describe by applying game levels [24]. This study discovered that it is possible to compose animations of a single movement with other movements to be appreciated as a unified dance choreography.

Computational Perspective: According to Luo, et al. [34], perspective expression can be achieved by preparing tasks in the form of codes and logically connecting blocks. However, the participant questions based on the interview results can explore the three computational perspectives. The expression perspective is investigated by examining how participants express their emotions.

“Having the ability to create animations is a plus. Which has never existed before. Never made” (Participant 4). Based on the excerpts from these responses, the expression of perspective can be demonstrated by how students feel after completing their project. These results are consistent with the initial definition of expressing an opinion, which is to use technology for consumption and to communicate ideas creatively.

This study examined the perspective of connecting related access to others through project-related assistance. Participant number 5 remarked, “It is beneficial. This appears to be a block that is disconnected from top to bottom. Now, with the assistance of the quirks” (Participant 5). According to these responses, assistance from others also plays a role in the connecting perspective. This is consistent with Falloon [35]'s assertion that connecting perspectives is a common technique utilised at all stages of student work.

The questioning perspective is condensed by asking what contribution you intend to make. Participant 1 provided a fascinating response “To create a public service announcement. Because they frequently participate in it. Participation is comparable to activities without plastic packaging. Therefore, I decided to protect the environment”. Developing students' ability to solve a problem in their environment can be facilitated by fostering curiosity about what contribution can be made through animation. The results of this study concur with Falloon [35]'s findings that questioning is also a component of students' collaborative approach to developing solutions.

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

The Influence of Computational Thinking

The analysis of the three aspects of computing was used to gain insight into the impact of Computational Thinking on problem-solving for students in grades K-12. The scope of problem-solving in this study includes how students complete their animation projects and what contributions they will make to the community.

Troubleshooting assignment

Midway through the project development process, students begin to struggle. As explained in the preceding section, testing & debugging, reusing & remixing play a crucial role in resolving this issue. In addition, the computing perspective, particularly the connecting perspective, contributes to students' efforts to address their challenges. The relevant connecting view is that it requires assistance from other parties. As stated by Participant 3, it is challenging to select movements with the assistance of others.

The findings concur with Shen, et al. [36]‘s assertion that problem-solving requires a shift in Computational Thinking skills from one pattern to another. Thus, problem-solving in a given task requires applying specific Computable Thinking skills, such as testing and debugging, reusing and remixing, and connecting.

Resolve environment issues

Students' problem-solving skills can also be apparent in how they help individuals ar Students' problem-solving abilities can also be observed in the manner in which they assist peers and complete projects. This has a great deal to do with the capacity to question viewpoints. An intriguing response was provided by a student who brought up the issue of protecting the surrounding environment. He stated that animation could be used to create public service announcements "I frequently participate in a plastic-free lifestyle. Therefore, it occurred to us to protect the environment " (Participant 1).

The findings are consistent with Brennan and Resnick's explanation that, with a questioning attitude, young people do not perceive a disconnect between the surrounding technology and their ability to negotiate reality in their daily life [24].

CONCLUSION

Based on the research data analysis, it can be ascertained that Code.org supports the development of K–12 students' computational thinking (CT) and problem-solving abilities through visual programming activities. The “Dance Party” module had a notable impact from the beginning to the end of the animation project. Earlier video tutorials and exercises helped students initiate their projects, while the implementation of testing and debugging practices, as well as reusing and remixing, enabled them to overcome obstacles during the middle stages. In the final phase, students expressed their reflections and emotions, which aligned with the expressing perspective of the Brennan and Resnick framework.

The findings showed that six out of the seven computational concepts could be introduced through the “Dance Party” activity, and four computational practices were actively applied during project development. Additionally, the three computational perspectives—expressing, connecting, and questioning—were evident in students’ reflections, especially when relating their learning experience to broader social or environmental issues.

However, these results must be considered within the limitations of the study. The small, purposively selected sample of five students and the short intervention duration limit the generalizability and depth of longitudinal insight. While the study prioritizes in-depth understanding over breadth, further research is needed to investigate how CT skills develop over time and transfer across contexts. Moreover, this study did not focus on measuring CT proficiency quantitatively, as the primary goal was to explore student experiences and meaning-making through a qualitative lens.

Future studies are encouraged to examine CT retention and progression through longitudinal designs, compare the effectiveness of different platforms such as Scratch or Tynker, and explore how visual programming tools can be adapted for low-resource educational settings. Additionally, involving a more demographically diverse sample would strengthen understanding of how different learners experience CT development.

Despite its scope and limitations, this study contributes valuable narrative insight into how K–12 students experience computational thinking through creative, student-centered programming activities—highlighting the importance of qualitative approaches in understanding the human aspect of learning to code.

ACKNOWLEDGMENT

Optional property

REFERENCES

[1] D. Menon, S. Bp, M. Romero, and T. Viéville, "Going beyond digital literacy to develop computational thinking in K-12 education," ed: Taylor&Francis (Routledge), 2019.

[2] S. Grover and R. Pea, "Computational thinking: A competency whose time has come," *Computer science education: Perspectives on teaching and learning in school,* vol. 19, no. 1, pp. 19-38, 2018.

[3] C. Wawan, K. Fenyvesi, A. Lathifah, and R. Ari, "Computational thinking development: Benefiting from educational robotics in STEM teaching," *European Journal of Educational Research,* vol. 11, no. 4, 2022.

[4] J. M. Wing, "Computational thinking benefits society," *40th anniversary blog of social issues in computing,* vol. 2014, p. 26, 2014.

[5] E. Relkin, L. E. de Ruiter, and M. U. Bers, "Learning to code and the acquisition of computational thinking by young children," *Computers & education,* vol. 169, p. 104222, 2021.

[6] S. Amri, C. W. Budiyanto, K. Fenyvesi, R. A. Yuana, and I. Widiastuti, "Educational Robotics: Evaluating the Role of Computational Thinking in Attaining 21st Century Skills," *Open Education Studies,* vol. 4, no. 1, pp. 322-338, 2022.

[7] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Computers in human behavior,* vol. 41, pp. 51-61, 2014.

[8] S. Çiftci and A. Bildiren, "The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children," *Computer science education,* vol. 30, no. 1, pp. 3-21, 2020.

[9] F. Kalelioglu, Y. Gulbahar, and V. Kukul, "A framework for computational thinking based on a systematic research review," 2016.

[10] J. Fagerlund, P. Häkkinen, M. Vesisenaho, and J. Viiri, "Computational thinking in programming with Scratch in primary schools: A systematic review," *Computer Applications in Engineering Education,* vol. 29, no. 1, pp. 12-28, 2021.

[11] S. Papadakis, M. Kalogiannakis, and N. Zaranis, "Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study," *International Journal of Mobile Learning and Organisation,* vol. 10, no. 3, pp. 187-202, 2016.

[12] D. Bau, J. Gray, C. Kelleher, J. Sheldon, and F. Turbak, "Learnable programming: blocks and beyond," *Communications of the ACM,* vol. 60, no. 6, pp. 72-80, 2017.

[13] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Computers & Education,* vol. 72, pp. 145-157, 2014.

[14] W. R. Sena Rivas, F. J. Herrero Gutiérrez, and S. Casillas Martín, "ICT-mediated education in youth and adult literacy programmes in the Dominican Republic: An approach to the state of the art," *Texto Livre: Linguagem e Tecnologia,* vol. 11, no. 3, pp. 131-153, 2018.

[15] M. U. Bers, "Coding and computational thinking in early childhood: The impact of ScratchJr in Europe," *European Journal of STEM Education,* vol. 3, no. 3, p. 8, 2018.

[16] E. Macrides, O. Miliou, and C. Angeli, "Programming in early childhood education: A systematic review," *International Journal of Child-Computer Interaction,* vol. 32, p. 100396, 2022.

[17] M. R. Fadhillah, C. W. Budiyanto, and P. Hatta, "The influence of block-based programming to computational thinking skills: A systematic review," in *AIP Conference Proceedings*, 2023, vol. 2540, no. 1: AIP Publishing.

[18] E. Skaraki and F. Kolokotronis, "Preschool and early primary school age children learning of computational thinking through the use of asynchronous learning environments in the age of Covid-19," *Advances in Mobile Learning Educational Research,* vol. 2, no. 1, pp. 180-186, 2022.

[19] K. Louka, "Programming environments for the development of CT in preschool education: A systematic literature review," *Advances in Mobile Learning Educational Research,* vol. 3, no. 1, pp. 525-540, 2023.

[20] K. Dilmen, S. B. Kert, and T. Uğraş, "Children's coding experiences in a block-based coding environment: a usability study on code. org," *Education and Information Technologies,* pp. 1-26, 2023.

[21] B. Arfé, T. Vardanega, and L. Ronconi, "The effects of coding on children's planning and inhibition skills," *Computers & Education,* vol. 148, p. 103807, 2020.

[22] D. Lambić, B. Đorić, and S. Ivakić, "Investigating the effect of the use of code. org on younger elementary school students' attitudes towards programming," *Behaviour & Information Technology,* vol. 40, no. 16, pp. 1784-1795, 2021.
[23] Y. I. Maharani, C. W. Budiyanto, and R. A. Yuana, "The art of computational thinking through visual programming: A literature review," in *AIP Conference Proceedings*, 2023, vol. 2540, no. 1: AIP Publishing.
[24] K. Brennan and M. Resnick, "Using artifact-based interviews to study the development of computational thinking in interactive media design," in *annual American Educational Research Association meeting, Vancouver, BC, Canada*, 2012, pp. 1-25.
[25] M. Guenaga, A. Eguluz, P. Garaizar, and J. Gibaja, "How do students develop computational thinking? Assessing early programmers in a maze-based online game," Computer Science Education, vol. 31, no. 3, pp. 279–305, 2021, doi: 10.1080/08993408.2021.1903248.
[26] R. K. Yin, *Case study research and applications*. Sage, 2018.
[27] D. Ary, L. C. Jacobs, C. K. S. Irvine, and D. Walker, *Introduction to research in education*. Cengage Learning, 2018.
[28] I. Etikan, S. A. Musa, and R. S. Alkassim, "Comparison of convenience sampling and purposive sampling," *American journal of theoretical and applied statistics,* vol. 5, no. 1, pp. 1-4, 2016.
[28] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology,* vol. 3, no. 2, pp. 77-101, 2006/01/01 2006.
[30] M. Maguire and B. Delahunt, "Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars," *All Ireland Journal of Higher Education,* vol. 9, no. 3, 2017.
[31] P. Liamputtong, "Handbook of research methods in health social sciences," 2019.
[32] S. Zha, Y. Jin, P. Moore, and J. Gaston, "Hopscotch into coding: introducing pre-service teachers computational thinking," *TechTrends,* vol. 64, pp. 17-28, 2020.
[33] B. K. Litts, W. E. Lewis, and C. K. Mortensen, "Engaging youth in computational thinking practices through designing place-based mobile games about local issues," *Interactive Learning Environments,* vol. 28, no. 3, pp. 302-315, 2020.
[34] F. Luo, P. D. Antonenko, and E. C. Davis, "Exploring the evolution of two girls' conceptions and practices in computational thinking in science," *Computers & Education,* vol. 146, p. 103759, 2020.
[35] G. Falloon, "An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad," *Journal of Computer Assisted Learning,* vol. 32, no. 6, pp. 576-593, 2016.
[36] J. Shen, G. Chen, L. Barth-Cohen, S. Jiang, and M. Eltoukhy, "Connecting computational thinking in everyday reasoning and programming for elementary school students," *Journal of Research on Technology in Education,* vol. 54, no. 2, pp. 205-225, 2022.