Effects of Computational Participation-Based Programming Education on Communication and Collaboration Skills of Gifted Students

Seung-Mee Lee and Seok-Ju Chun*

Department of Computer Education, Seoul National University of Education, Seoul, Republic of Korea
Email: 1226me@naver.com (S.-M.L.); chunsj@snue.ac.kr (S.-J.C.)
*Corresponding author

Manuscript received June 9, 2025; revised June 30, 2025; accepted July 31, 2025; published December 9, 2025

***Abstract*—This study examines the effects of a programming education program based on computational participation on the communication and collaboration skills of gifted elementary students. While conventional computing education for gifted students primarily focused on individual cognitive and technical skill development, the program was explicitly designed to foster personal expression, peer collaboration, and community engagement through programming. Over the course of 16 class periods, students engaged in a sequence of individual, remix, and collaborative programming projects using the Delightex platform. To evaluate its impact, pre- and post-program assessments of communication and collaboration skills were conducted, complemented by student interviews. The results revealed an average increase in students' communication skill scores, though the change was not statistically significant. Collaboration skill scores, however, demonstrated a statistically significant improvement. These findings indicate that computational participation-based programming has the potential to support the development of collaboration and communication skills. These competencies are recognized as essential for the 21st century and can be nurtured within a socially interactive learning environment for gifted students.**

***Keywords*—collaboration skills, communication skills, computational participation, gifted students, programming education**

I. Introduction

The rapid advancement of digital technologies requires students to develop strong technological competencies to adapt to today's globalized world [1]. Accordingly, countries around the world are striving to enhance national competitiveness by identifying and nurturing talented individuals, particularly in the Software (SW) industry. In this context, fostering creative talents capable of leading the era of digital transformation and Artificial Intelligence (AI) has emerged as a key national priority.

Against this backdrop, Computational Thinking (CT) has been emphasized as a core concept in computing education and as a mindset applicable to various domains, including problem solving, system design, and understanding human behavior [2]. Traditionally, CT-based programming education has focused on fostering self-directed learning by engaging students in iterative processes of exploration and problem solving. However, most CT-based educational programs tend to concentrate on cognitive and technical competencies, such as logical reasoning, problem-solving, and algorithm design, while often overlooking students' socio-emotional development [3].

As the limitations of traditional CT-based programming education have become increasingly evident, there is growing recognition of the need for approaches that extend beyond individual cognitive skills to also support learners' social participation and creative collaboration. In this context, the concept of "computational participation" has emerged as an expanded framework. It encourages learners to express their ideas through programming, remix the work of others, and engage in meaningful interactions within a community [4]. This approach suggests the potential of programming education to foster essential 21st-century skills, particularly Creativity, Critical thinking, Collaboration, and Communication (4C skills).

In the era of the knowledge-based society and the Fourth Industrial Revolution, students are increasingly expected to acquire such 4C skills [5]. Education in the 21st century carries the important responsibility of nurturing not only students' character and basic literacy but also their ability to think critically, collaborate effectively, communicate clearly, and demonstrate creativity. By developing these skills, students can actively participate in collaborative problem-solving in their daily lives, respect differing opinions, and cultivate innovative thinking [6].

In response to this need, a variety of educational initiatives have been implemented to enhance students' 4C skills [7–9]. Among them, block-based programming environments have proven particularly effective in promoting 4C skill development, as they allow learners to visually express their thinking and receive immediate feedback [10]. Furthermore, when students engage in programming with and for others, their communication and collaboration skills are significantly enhanced [11]. For this reason, it is important for educators to intentionally design programming activities that promote active social interaction among students during the learning process [11].

Based on this perspective, this study aims to examine the effects of a computational participation-based programming education program on the communication and collaboration skills of gifted elementary students in computing. The program was designed based on the core elements of computational participation, which include personal expression, peer collaboration, and community engagement. This study analyzed the changes in students' communication and collaboration skills before and after participating in the program. Through this, the study seeks to explore the educational potential of computational participation-based programming education in fostering key 21st-century skills.

II. Literature Review

Computational participation is a participation-centered learning paradigm in which learners do not simply acquire programming skills or Computational Thinking (CT), but engage in meaningful creation, connection, and contribution within social contexts through the medium of technology [12, 13]. Based particularly on Papert's constructionism, computational participation emphasizes that students engage in programming learning that is connected to personal, social, and cultural contexts through computing activities [14, 15]. This perspective aligns with Kafai and Burke's concept of the "social turn" in K–12 computing education [13], which is embodied in three shifts: from coding to application creation, from original creation to remixing, and from tool design to community-based participation.

Kafai and Burke [14] argued that programming education should not be limited to syntax learning but should be a practice where learners create and share meaningful outcomes, such as games or interactive stories, within social contexts and communities, placing greater emphasis on participation than on the tool itself. Ultimately, computational participation suggests that learners' digital activities should extend beyond simple problem-solving to express their identities and interests (personal aspect), interact with others (social aspect), and establish meaningful connections with broader communities (cultural aspect).

At the activity level, the personal aspect of computational participation involves learners expressing their ideas, interests, thoughts, and identities through programming to create meaningful outcomes [14]. Students create narratives, games, and interactive works through programming, enabling them to visualize and externalize their thinking. Such activities encourage students to see technology not merely as a problem-solving tool but as a medium for self-expression. Burke *et al*. [4] described this as "programming as a new genre of composition", viewing programming as an extension of writing for self-expression.

The social aspect of computational participation involves collaborative practices where learners plan and design together, exchange ideas during problem-solving, and co-create shared outcomes [4]. Kafai [12] referred to this as "participation in digital activities", highlighting that computing learning should not be an isolated activity but a participatory practice involving others. Fields *et al*. [16], in their analysis of Scratch community activities, explained that through remixing, commenting, and exchanging feedback, learners naturally engage in collaboration and community-based creation.

The cultural aspect of computational participation involves learners actively contributing to digital communities by sharing their work, remixing or reinterpreting others' creations, and engaging in activities that promote public value and social meaning [14]. This signifies that programming should move beyond individual accomplishment to socially connected activities. Kafai and Proctor [15] emphasized that computing serves as a means of connecting with others and shaping one's identity. Furthermore, Kafai and Burke [13] identified remixing others' works as a core practice of computational participation and argued that programming should be conducted on the foundation of community participation.

Empirical studies of computational participation practices include the following. Sivaraj *et al.* [17] analyzed game-based learning and citizen science projects using platforms such as Whyville and WeatherBlur, illustrating how learners engaged in community-based problem-solving through technology, thereby enacting computational participation. This study demonstrated that computational participation can extend STEM learning into a practical learning experience that includes collaboration, identity formation, and community contribution. Kuo and Hsu [18] found that in CT-focused board game lessons, learners achieved computational participation by sharing ideas, collaborating, and working toward common goals even without coding. Additionally, Flynn *et al*. [19], in a longitudinal study integrating computational participation into elementary mathematics education, reported that computational participation enhanced students' collaboration, problem-solving, perseverance, and engagement, while also positively influencing teachers' instructional practices and promoting interdisciplinary thinking.

Collaborative programming has been shown to enhance students' problem-solving skills by encouraging them to observe peers' code, learn from mistakes, and reflect metacognitively. It also fosters motivation and positive attitudes, as students iteratively improve their work and verbalize their ideas more clearly throughout the process [20, 21]. Especially, real-time collaboration platforms have been found to support not only peer learning of CT but also mutual communication and co-creation among learners [22]. Furthermore, Shahin *et al.* [23] found that collaborative practices were the most accessible aspects of computational thinking for secondary school girls, suggesting that future CT and computer science programs for girls should actively emphasize and support collaboration.

Based on the concept of computational participation, this study designs and implements a programming class that emphasizes collaborative engagement. In particular, the class utilizes Delightex, an educational platform that enables the integrated design and programming of 3D virtual spaces. Delightex supports real-time collaborative programming by allowing multiple users to access the same project simultaneously and co-design the same project, making it possible to engage in synchronous collaboration beyond physical constraints.

III. Methods

A. Research Design

This study was conducted with elementary school students identified as gifted in computing. The purpose of the study was to analyze students' participation patterns and learning experiences, and to examine the development of their communication and collaboration skills in their programming class. The programming education program was designed based on the three core elements of computational participation—personal expression, collaboration, and community involvement. This study employed a single-group pretest-posttest design, a quasi-experimental quantitative design without a control group. In addition to the quantitative data collected through pre- and post-surveys, student interviews were conducted to provide supplementary insights into the participants' learning experiences and to help interpret the changes observed in the survey results. The overall structure of data collection and instructional flow is summarized in Table 1. A self-assessment scale measuring students' communication and collaboration skills was administered both before the first session (pre-test) and after the final session (post-test) to assess learning outcomes. In addition, semi-structured student interviews were conducted at four points during the program, following each 4-session block, to supplement and interpret the quantitative findings.

Table 1. Data collection and instructional timeline

Phase	Instrument	Timing
Pre-test	Communication and collaboration test	Before 1st session
Intervention	Computational participation-based programming education program (16 sessions)	
Interviews	Semi-structured student interviews	After every 4th session (4 times total)
Post-test	Communication and collaboration scale	After the final 16th session

The participants were 19 fifth-grade students (16 boys and 3 girls) enrolled in an elementary computing gifted education center affiliated with a university. All participants were selected through an entrance examination and, therefore, represented a group with comparable levels of programming proficiency. The program was conducted once a week, with each session consisting of four class periods, for a total of four weeks (16 class periods). After each session, students participated in an interview reflecting on their learning activities. This study followed ethical guidelines for research involving minors. Before participation, written informed consent was obtained from all students and their legal guardians. Participants were informed of the voluntary nature of the study and assured that their data would remain confidential and would be used solely for research purposes.

The participating students used the Delightex platform to construct 3D virtual environments, perform programming tasks, and engage in collaborative activities, including sharing and remixing digital content. Delightex is an educational platform that integrates programming and design within a 3D virtual environment, allowing learners to create interactive projects by combining visual and programming elements. Notably, the platform supports real-time collaborative programming, allowing multiple users to simultaneously access and co-develop a shared project in a 3D space, thereby overcoming physical limitations.

In addition, Delightex allows users to share their projects with options that permit others to modify and expand upon them, enabling remix activities at both the design and code levels. These features facilitate experiences where learners can creatively build upon or enhance others' work, fostering not only their programming abilities but also their collaboration and creativity skills.

B. Education Program

The programming education based on the computational participation framework was structured over a total of 16 class periods. Using the Delightex platform, students created 3D virtual spaces and engaged in a variety of programming activities within these spaces.

Students sequentially experienced individual projects, remix activities, and team projects, completing four major projects: "Creating My Virtual Exhibition", "Creating My Own Escape Room", "Escape Room Remix", and "Team Project".

Each stage of the program was carefully designed to integrate the three core elements of computational participation—personal expression, collaboration, and community participation. In particular, personal expression was fostered throughout all project stages, as students expressed their unique ideas through 3D space design and programming. Collaboration was emphasized in the "Escape Room Remix" project, where students built upon and enhanced their peers' projects through remixing. The "Team Project" project further promoted collaboration by having students co-develop projects from the initial planning phase. Community participation was integrated into Sessions 2, 5, 8, and 14–16, where students shared their projects with peers, presented their work, and received peer feedback through various sharing and presentation activities.

The appropriateness and validity of the program were reviewed and validated by two experts holding doctoral degrees in computer education. The detailed instructional content is presented in Table 2, and examples of student-created project work during the classes are shown in Figs. 1 and 2. Fig. 3 shows students actively engaged in designing 3D virtual spaces and programming interactive elements using the Delightex.

Table 2. Contents of classes

Session	Learning activities
1	Explore sample projects in Delightex and learn basic functions
2	Create an virtual exhibition and share with peers
3	Learn basic programming features in Delightex
4	Design my escape room project
5	Share escape room projects via Padlet and explore peers' work
6–7	Remix and improve a peer's escape room game
8	Present the remixed project and share with the class
9	Form teams and decide on a project topic and plan
10–12	Conduct the Delightex team project
13	Finalize the Delightex team project
14–16	Present the Delightex team project, give peer evaluation, and feedback

Fig. 1. Example of a student-created project: Escape Room Remix.

Fig. 2. Example of a student-created project: Team project.

Fig. 3. Students engaging in programming with Delightex.

C. Instruments and Analysis

This study aimed to examine the effects of the educational program on two of the 4C skills: communication and collaboration. To this end, relevant prior studies on 4C skill assessment instruments were reviewed, and an instrument appropriate for the study participants was developed and administered before and after the program.

The instrument was designed to assess students' self-perceived communication and collaboration skills, as it was based on self-reported responses. All items were rated on a 5-point Likert scale. The instrument consisted of 14 items in total: 6 items measuring communication skills and 8 items measuring collaboration skills.

The internal consistency reliability of the instrument was found to be acceptable. For the communication domain, Cronbach's α was 0.776 for the pre-test and 0.755 for the post-test. For the collaboration domain, Cronbach's α was .799 for the pre-test and 0.777 for the post-test.

Given that the sample size was fewer than 30 participants, a normality test was conducted. The results of the Shapiro-Wilk test indicated that normality was satisfied, with $p = 0.130$ for communication and $p = 0.185$ for collaboration in the pre-test, and $p = 0.185$ for communication and $p = 0.136$ for collaboration in the post-test, all exceeding the significance level of 0.05. Subsequently, pre- and post-program comparisons were analyzed using SPSS 30.0.

IV. Results

The results of analyzing the effects of the computational participation-based programming education program for gifted elementary students are presented in Table 3.

In the area of communication, the mean score increased from 23.95 (pre-test) to 24.90 (post-test). However, the paired-samples t-test result ($t = –1.01$, $p = 0.32$) indicated that this increase was not statistically significant. Based on the normalized gain (N-Gain) analysis, the communication domain showed a gain of 0.157, which is classified as low improvement.

In the area of collaboration, the mean score increased significantly from 32.37 (pre-test) to 35.47 (post-test), with the paired-samples t-test yielding $t = –2.54$, $p = 0.02$*. This indicates a statistically significant improvement and demonstrates that the computational participation-based programming education positively influenced the development of students' collaboration skills. According to the N-Gain analysis, the collaboration domain showed a gain of 0.406, indicating a medium level of improvement. As classified by Hake [24], N-Gain values below 0.3 are considered low, values between 0.3 and 0.69 are medium, and values of 0.7 or above are high.

Table 3. Pre-post difference analysis

Area	Pre (N=19)		Post (N=19)		T	p	N-Gain
	M	SD	M	SD			
Communication	23.95	3.17	24.90	3.53	–1.01	0.32	0.157
Collaboration	32.37	4.32	35.5	3.52	–2.54	0.02^{*}	0.406

$^{*}p < 0.05$

The pre- and post-intervention changes in both communication and collaboration skills are visually illustrated in Fig. 4.

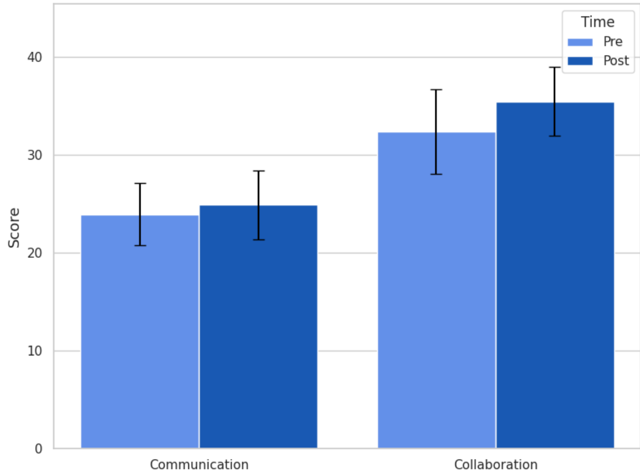

Fig. 4. Pre-post changes in communication and collaboration.

This improvement in collaboration skills, as shown in Fig. 3, can be attributed to the program's focus on collaborative learning activities. Opportunities to remix peers' projects and participate in team-based programming projects appear to have contributed effectively to fostering students' ability to work collaboratively.

These results suggest that the learning environment designed with the principles of computational participation, which emphasizes interaction, co-creation, and community engagement, helped enhance the collaboration skills of gifted elementary students.

However, the limited improvement in communication may be attributed to the nature of the activity design, which focused more on co-creation and code sharing than on structured verbal interaction. In particular, the remix activity, in which students modified and extended their peers' projects, encouraged creative collaboration but required minimal verbal exchange. While the collaborative nature of the group projects and remix tasks was expected to naturally promote peer communication, such interactions appeared to be less effective than anticipated in fostering significant improvements in communication skills.

To gain a deeper understanding of how students experienced and perceived computational participation during the program, a qualitative analysis of their interview responses was conducted. Students participated in interviews at the end of each four-session block, reflecting on their learning experiences related to computational participation. The interview data were systematically organized, key units of meaning were identified and coded, and these codes were then grouped according to similar characteristics or meanings. The analysis was structured around the three elements of computational participation (personal expression, collaboration, and community involvement). The key themes identified from these responses are summarized in Table 4.

Table 4. Student interview

Session	Interview
1–4	- I was really happy that I got to build my world and code everything just the way I wanted. - Watching how others coded made me want to try doing it like them. - Sharing my project made me reflect on how I could improve it and what needed to be fixed. - After hearing feedback from my friends, I felt I should revise my project and make it better. - Even if the feedback wasn't all positive, I was really happy that someone viewed my project and shared their thoughts.
5–8	- I liked how I rewrote the text, added various characters and objects, and customized the project to match my ideas. - We told each other what parts of our games were good and what could be better, and then edited each other's games to make new versions. - It was fun to take my friend's code and add my ideas to it. - I felt proud because my project got a lot of hearts, even though I knew there were still things to fix. - It was fun, and I found it helpful to see what others did so I could use those ideas in my project. - I learned what I did well, what I could improve, and what others were good at, which helped me apply that to my project. - After seeing others' work, I realized some people had better ideas, and I was able to use those ideas in my own game.
9–12	- We helped each other figure things out when we didn't know what to do. - My friends helped me with coding the next scene when I ran out of time because I had made things too complicated. - One of my teammates had a smart idea that made our code simpler. - We shared our ideas and helped each other with the coding.
13–16	- I felt most proud when I wrote the code myself and improved the quality of the final result. - Next time, I want to follow what my friends did and use that to make our game even better. - I felt amazed and excited to see new ideas I hadn't thought of before. - I came up with even more ideas I could use for my next project.

An analysis of students' interview responses revealed that they perceived collaborative experiences as valuable opportunities to compensate for their limitations, solve problems together, and enhance their projects with peers. Statements indicating a sense of pride and satisfaction when incorporating peer feedback or seeing their projects remixed and improved suggest that such collaborative experiences positively influenced their learning motivation.

Students actively engaged in both communication and collaboration at the code level by referencing peers' code to add new features and by restructuring existing projects to create new games. They also demonstrated effective communication and teamwork by providing mutual support, distributing roles autonomously, and working toward shared goals throughout the collaborative process.

Through these experiences, students recognized that collaboration enabled them to overcome challenges and co-create meaningful outcomes with their peers, which they regarded as enjoyable and enriching learning experiences. This finding highlights the potential of programming activities to evolve from individual tasks into socially interactive, co-creative processes.

Notably, students expressed joy and pride when their projects inspired others or when they applied impressive ideas observed in their peers' creations. Several students also acknowledged that remixing by peers enhanced the quality of their work. In addition, many students reported applying new ideas to their projects after observing the creative approaches of their classmates, demonstrating an active role as both contributors and participants in the learning community.

Importantly, students indicated greater engagement and motivation when their projects were shared with others and received feedback from new users, compared to when they programmed individually. These findings suggest that programming activities, when embedded in a collaborative and interactive learning community, can significantly foster both communication and collaboration skills.

V. Conclusion

This study applied a programming class designed from the perspective of computational participation to elementary gifted students in computing and examined the changes in their communication and collaboration skills. Unlike traditional gifted education in computing, which tends to focus on individual learning, this study structured learning activities that encouraged students to express their ideas, interact with peers, and experience meaningful creation within a community, thereby fostering communication and collaboration skills.

The results showed that while communication skills exhibited an increase in average scores, the change was not statistically significant. Collaboration skills showed significant improvement. Analysis of student interviews revealed that students actively engaged in collaborative practices by sharing and remixing code, asking questions, and providing advice to peers. These interactions also had a positive impact on their communication experiences. Furthermore, receiving positive responses (such as likes, comments, and remixes) on their projects helped students develop a sense of belonging within the learning community and foster self-efficacy and motivation by recognizing that their work could influence others.

These findings demonstrate that programming learning can move beyond technical training into a collaborative and communication-centered creative activity within a social context. This interpretation is also consistent with previous studies that implemented computational participation through various forms of educational activities. In line with these studies, the present research supports the view that programming education based on computational participation can promote collaboration, peer engagement, and community-based learning. However, while the use of a real-time collaborative platform in this study appeared to play a meaningful role in promoting students' active participation and co-creation, the results did not show a statistically significant improvement in communication skills. This contrasts with previous studies that reported real-time collaboration as a key factor in facilitating peer communication and interaction.

Therefore, gifted education in computing should move beyond basic coding instruction to incorporate collaborative programming experiences that actively nurture students' communication and collaboration skills. This study suggests that a computational participation-based programming education approach can offer meaningful educational experiences that contribute to the development of these critical skills among gifted students, providing important implications for the future direction of gifted education and instructional design.

This study has several limitations. First, it measured students' communication and collaboration skills using a self-reported instrument, which may be subject to response bias and may not fully reflect actual performance. While the findings indicate positive changes in perceived competencies, future research should incorporate performance-based assessments or observational measures to strengthen the validity of the results. Second, the study was conducted with a relatively small group of 19 elementary students identified as gifted in computing. Therefore, the findings should be interpreted with caution and may not be generalizable to the broader student population. Furthermore, because the participants demonstrated exceptionally high levels of motivation and interest in programming, these characteristics may have amplified the positive effects of the program. Future research should include larger sample sizes and examine whether programming education based on computational participation is also effective for students with diverse backgrounds and varying levels of prior experience and interest.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Mrs. Lee researched and implemented the computational participation-based coding education program, conducted the classroom instruction, and collected and analyzed survey data from the participants. Dr. Chun conceptualized the research framework and contributed to the design of the education program. Dr. Chun and Mrs. Lee collaboratively analyzed the findings and co-wrote the manuscript. All authors critically revised the manuscript for important intellectual content and approved the final version for publication.

References

[1] I. Irham, I. Tolla, and B. Jabu, "Development of the 4C teaching model to improve students' mathematical critical thinking skills," *International Journal of Educational Methodology*, vol. 8, no. 3, pp. 493–504, 2022. doi: 10.12973/ijem.8.3.493

[2] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[3] G. K. W. Wong, B. Y. P. Tsang, Q. Wu, and X. Zhang, "Do underprivileged youth find hope, sense of community, and perceived social support in computational participation? A socio-cognitive approach to computational learning," *Education and Information Technologies*, vol. 28, pp. 8975–8997, 2023. doi: 10.1007/s10639-022-11522-6

[4] Q. Burke, W. I. O'Byrne, and Y. B. Kafai, "Computational participation: Understanding coding as an extension of literacy instruction," *Journal of Adolescent & Adult Literacy*, vol. 59, no. 4, pp. 371–375, 2016. doi: 10.1002/jaal.496

[5] M. A. Masruri, A. Efendi, and S. Sumaryati, "Innovative project-based learning video tutorial media: Development and its effect on students' collaborative skills," *Jurnal Teknologi Pendidikan*, vol. 26, no. 3, pp. 926–943, 2024. doi: 10.21009/JTP2001.6

[6] R. Rudianto, R. Diani, S. Subandi, and N. Widiawati, "Development of assessment instruments 4C skills (critical thinking, collaboration, communication, and creativity) on parabolic motion materials," *Journal of Advanced Sciences and Mathematics Education*, vol. 2, no. 2, pp. 65–79, 2022. doi: 10.58524/jasme.v2i2.115

[7] N. Vebriani, Usmeldi, D. Irfan, and Ta'ali, "Development of a STEM-based mathematics e-module to improve 4C skills of vocational students," *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran*, vol. 10, no. 2, pp. 510–521, 2024.

[8] Kurniahtunnisa and E. C. Wowor, "Development of STEM-project based learning devices to train 4C skills of students," *Bioeduca: Journal of Biology Education*, vol. 5, no. 1, pp. 66–78, 2023.

[9] J. Leonard, C. Djonko-Moore, K. R. Francis, A. S. Carey, M. B. Mitchell, and I. D. Goffney, "Promoting computational thinking, computational participation, and spatial reasoning with LEGO robotics," *Canadian Journal of Science, Mathematics and Technology Education*, vol. 23, pp. 120–144, 2023. doi: 10.1007/s42330-023-00267-0

[10] X. Weng, Z. Cui, O. L. Ng, M. S. Y. Jong, and T. K. F. Chiu, "Characterizing students' 4C skills development during problem-based digital making," *Journal of Science Education and Technology*, vol. 31, pp. 372–385, 2022. doi: 10.1007/s10956-022-09961-4

[11] B. Jiang, W. Zhao, X. Gu, and C. Yin, "Understanding the relationship between computational thinking and computational participation: A case study from Scratch online community," *Educational Technology Research and Development*, vol. 69, pp. 2399–2421, 2021. doi: 10.1007/s11423-021-10021-8

[12] Y. B. Kafai, "From computational thinking to computational participation in K–12 education," *Communications of the ACM*, vol. 59, no. 8, pp. 26–27, 2016. doi: 10.1145/2955114

[13] Y. B. Kafai and Q. Burke, "The social turn in K-12 programming: Moving from computational thinking to computational participation," in *Proc. the 44th ACM Technical Symposium on Computer Science Education*, pp. 603–608, 2013. doi: 10.1145/2445196.2445373

[14] Y. B. Kafai and Q. Burke, "Mindstorms 2.0: Children, programming, and computational participation," in *Proc. Constructionism 2014 Conference*, 2014, pp. 1–10.

[15] Y. B. Kafai and C. Proctor, "A revaluation of computational thinking in K–12 education: Moving toward computational literacies," *Educational Researcher*, vol. 51, no. 2, pp. 146–151, 2022. doi: 10.3102/0013189X211057904

[16] D. A. Fields, Y. B. Kafai, and M. T. Giang, "Youth computational participation in the wild: Understanding experience and equity in participating and programming in the online Scratch community," *ACM Transactions on Computing Education*, vol. 17, no. 3, 15, 2017. doi: 10.1145/3123815

[17] R. Sivaraj, J. A. Ellis, J. R. Wieselmann, and G. H. Roehrig, "Computational participation and the learner-technology pairing in K-12 STEM education," *Human Behavior and Emerging Technologies*, vol. 2, no. 4, pp. 387–400, 2020. doi: 10.1002/hbe2.220

[18] W. C. Kuo and T. C. Hsu, "Learning computational thinking without a computer: How computational participation happens in a computational thinking board game," *Asia-Pacific Education Researcher*, vol. 29, no. 1, pp. 67–83, 2020. doi: 10.1007/s40299-019-00479-9

[19] D. Flynn, J. Hughes, and J. A. Robb, "Improving mathematics learning through computational participation," *Journal of Educational Informatics*, vol. 4, no. 1, pp. 19–33, 2023. doi: 10.51357/jei.v4i1.214

[20] Krismadinata, Efan, C. Boudia, J. Jama, and A. Y. Saputra, "Effect of collaborative programming on students' achievement learning object-oriented programming course," *International Journal of Information and Education Technology*, vol. 13, no. 5, pp. 437–444, 2023. doi: 10.18178/ijiet.2023.13.5.1869

[21] X. Lai and G. K. W. Wong, "Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis," *British Journal of Educational Technology*, vol. 53, pp. 150–170, 2022. doi: 10.1111/bjet.13157

[22] T. C. Hsu, H. Abelson, E. Patton, S. C. Chen, and H. N. Chang, "Self-efficacy and behavior patterns of learners using a real-time collaboration system developed for group programming," *International Journal of Computer-Supported Collaborative Learning*, vol. 16, pp. 559–582, 2021. doi: 10.1007/s11412-021-09357-3

[23] M. Shahin, C. Gonsalvez, J. Whittle, C. Chen, L. Li, and X. Xia, "How secondary school girls perceive computational thinking practices through collaborative programming with the micro:bit," *Journal of Systems and Software*, vol. 183, 111107, 2022. doi: 10.1016/j.jss.2021.111107

[24] R. R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *American Journal of Physics*, vol. 66, no. 1, pp. 64–74, 1998. doi: 10.1119/1.18809