IELTech Informatics Education and Learning Technologies

Vol 1, No. 1, June 2025, pp. 20–30
doi: 10.23917/ieltech.v1i1.16524

Improving Reasoning Skills in Primary School Students: An Experimental Study on 2D Game Development with Scratch

Bima Samudra Wicaksana[1]*, Ahmad Apriyanto[2]

[1,2]Informatics Engineering Education, Universitas Muhammadiyah Surakarta, Indonesia
*Email: bimasamudraw@gmail.com

Article History:

Received: 13 February 2025
Revised: 28 April 2025
Accepted: 25 May 2025
Published: 4 June 2025

Keywords:

reasoning skills, Scratch, visual programming, project-based learning, 2D game development, elementary education, quasi-experimental

Abstract

Reasoning ability is a fundamental skill for problem-solving and decision-making, yet its development in elementary school computer education remains suboptimal, often limited to basic operational tasks. This study aims to investigate the effect of Scratch visual programming, implemented through 2D game development projects with the Project-Based Learning (PjBL) model, on the reasoning skills of fifth-grade students. A quasi-experimental one-group pretest-posttest design was employed, involving 30 students (14 male, 16 female) from a public elementary school in Wonogiri, Indonesia. The intervention consisted of five meetings where students collaboratively created drag-and-drop and jumping games using Scratch, following the PjBL syntax. Data were collected through a validated reasoning test adapted from Raven's Coloured Progressive Matrices (8 items, Cronbach's $\alpha = 0.634$) and the System Usability Scale (SUS) questionnaire. Expert validation confirmed the module's high validity (Aiken's $V = 0.70\text{-}0.76$) and reliability ($\alpha = 0.944\text{-}0.992$). The results revealed a statistically significant increase in reasoning scores from pretest ($M = 61.00$, $SD = 12.45$) to posttest ($M = 86.67$, $SD = 8.92$), with $t(29) = -8.19$, $p < 0.001$, and a large effect size (Cohen's $d = 1.50$). The SUS scores ($M = 86.8$) indicated high usability and acceptability of the learning method and module. Student project performance was excellent ($M = 95$), demonstrating successful application of logical sequencing and systematic planning. These findings provide empirical evidence that integrating Scratch visual programming with PjBL through 2D game development significantly enhances elementary students' reasoning abilities while offering a practical and engaging approach for computer education. The study contributes to the growing body of literature on block-based programming and computational thinking in primary education, with implications for curriculum design and instructional practice in the digital age.

1 Introduction

Reasoning ability, defined as the cognitive activity of connecting existing knowledge with new problem situations to form arguments or conclusions, is a crucial skill for human problem-solving and decision-making (Olsson & Granberg, 2024a). It involves logical thinking necessary to draw conclusions based on facts and build justifications for a position. The capacity for reasoning enables individuals to assess situations rationally, apply logical systems to available information, and ultimately support their future success in navigating complex global challenges. Consequently, fostering this skill is a paramount objective in 21st-century education, especially given the rapid digitalization of various aspects of life, including the educational field (Siswandi et al., 2024a; Sri Utami, 2024).

However, the reality in many elementary schools indicates that students' reasoning abilities, particularly within computer extracurricular activities, are often underdeveloped. Observations conducted at an elementary school in Wonogiri, Indonesia, revealed that traditional computer instruction typically focuses merely on basic operational tasks, such as turning the computer on and off, typing practice, and managing applications, thus limiting opportunities for students to engage in activities that effectively train their reasoning and computational thinking. This gap highlights the suboptimal implementation of computer education, failing to harness its potential as a medium for nurturing higher-order thinking skills, which are fundamental to navigating the challenges of the digital age. This concern is consistent with broader findings indicating that critical and logical thinking skills among Indonesian students remain relatively not so good, as reflected in international assessments such as PISA (Katoningsih & Sunaryo, 2020).

One promising and engaging strategy to address this gap is the incorporation of visual programming, specifically through the *Scratch* platform. *Scratch* is recognized as a fun, interactive visual programming language that enables novices, particularly elementary students, to create programs, stories, and animations without the complexities associated with syntax errors often found in conventional programming languages (Apriyanto & Sukirman, 2024; Romero & Artal-Sevil, 2021). By employing a block-based system, Scratch simplifies the process of coding, making it an effective medium for training reasoning skills as students are required to logically arrange code blocks to achieve desired outcomes (Koray & Bilgin, 2023). This ease of use and visual appeal also fosters student activity and creativity during learning. Recent studies have confirmed that Scratch not only provides a platform for learning programming but also supports the development of concrete operational cognitive skills, particularly for children in the 7-11 years age range (Stamatios, 2024). Furthermore, the interactive process of coding and receiving immediate visual feedback in the Scratch environment engages students in an iterative process of constructing solutions intertwined with emerging understanding (Zacharis, 2025).

To maximize the impact of Scratch on students' reasoning, this study integrates the visual programming tool with the Project-Based Learning (PjBL) model. PjBL is an innovative teaching technique that centers learning around complex problems or projects, requiring students to learn, solve problems, and ultimately produce tangible work or products (Monge Roffarello & Sáenz, 2025; Siswandi et al., 2024). This method not only enhances critical thinking and problem-solving capabilities but also boosts student motivation and group collaboration. The integration of PjBL with Scratch has been shown to have significant positive impacts on student learning experiences and outcomes, as it combines collaborative, hands-on learning with creative expression (Ming, Fuad, Lee, Yaacob, et al., 2025). In this context, the students' project involved the creation of functional 2D games, such as ”drag and drop” and ”jumping games,” a process which inherently demands systematic planning, logical sequencing, and continuous problem-solving, all crucial components of reasoning. Through such project-based activities, students can better apply logical concepts in solving problems via game creation in Scratch (Topalli & Cagiltay, 2018). The module developed for this purpose, proven valid and reliable by expert judgment, served as the foundation for the five-meeting intervention.

Therefore, this study employs a quasi-experimental, one-group pretest-posttest design to rigorously investigate the effect of using Scratch Visual Programming, implemented through 2D game development projects, on the reasoning abilities of elementary students. This design aligns with methodological approaches used in similar educational technology research examining interventions' effects on computational thinking and reasoning skills (Gao et al., 2025). The primary objective is to test the hypothesis that visual *Scratch* programming significantly improves the reasoning skills of primary school students. The findings are anticipated to provide empirical evidence supporting the use of constructive and project-based learning strategies in technology education to better prepare students for the demands of the 21st century.

2 Method

2.1 Research Design and Participants

This study employed a quantitative approach with a quasi-experimental design, specifically the One-Group Pretest-Posttest Design. This design was selected due to the impracticality of implementing full randomization in the school setting, a common constraint in educational research (Maggin, 2022). In this design, the dependent variable—students' reasoning ability—was measured twice: before the intervention (pretest, denoted as O1) and after the intervention (posttest, denoted as O2). The treatment (X) administered to the experimental group consisted of learning activities using PjBL model integrated with Scratch visual programming, supported by specifically developed learning modules. The research paradigm can be represented as O1 × O2, where the effect of the treatment is determined by comparing O2 and O1 .

The research was conducted at a public elementary school in Wonogiri Regency, Central Java, Indonesia, over a two-month period during the even semester of the 2024/2025 academic year. The participants comprised all fifth-grade students at the school, totaling 30 students (14 males and 16 females), as presented in Table 1. The selection of fifth-grade students was purposeful, considering that at this developmental stage (typically 10-11 years old), children are in the concrete operational stage of cognitive development, making them capable of logical thinking about concrete objects and events, which aligns with the cognitive demands of block-based programming.

Table 1: Demographic profile of research participants

Category	Description	Count
Participants	Male	14
	Female	16
Total		**30**

2.2 Research Procedure

The research procedure was systematically structured following the Analysis, Design, Development, Experiment, and Evaluation. Figure 1 illustrates the systematic flow of the research, adopting a rigorous methodology based on five interconnected stages. The initial Analysis phase involved comprehensive activities such as conducting a literature review and detailed document analysis, covering the existing syllabus, lesson plans (RPP), reference books, and the current curriculum. Following this, the Design phase focused on structuring the core components of the intervention, specifically the conceptualization of the PjBL module, the definition of learning topics, and the detailed planning of the quasi-experiment itself. This systematic pre-planning ensured that the research instruments and educational materials were fully aligned with the research objectives, which is a critical necessity for validating the feasibility of the subsequent intervention.

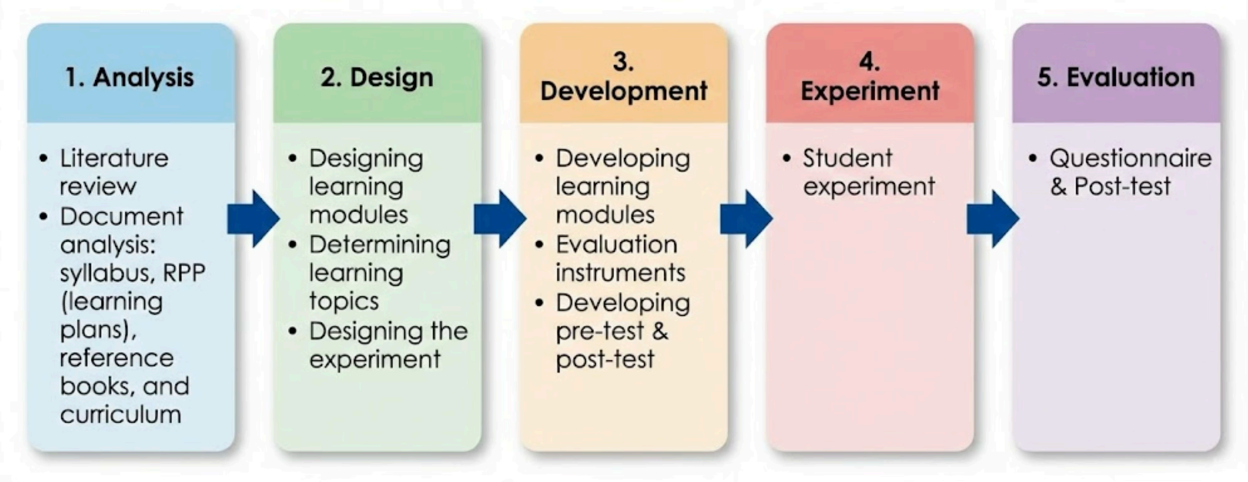
1. Analysis
• Literature review
• Document analysis: syllabus, RPP (learning plans), reference books, and curriculum
2. Design
• Designing learning modules
• Determining learning topics
• Designing the experiment
3. Development
• Developing learning modules
• Evaluation instruments
• Developing pre-test & post-test
4. Experiment
• Student experiment
5. Evaluation
• Questionnaire & Post-test

Figure 1: Research design flow diagram

The *Analysis* phase involved a comprehensive literature review on reasoning skills, computational thinking, and Scratch-based learning, complemented by document analysis of the school curriculum, syllabus, and existing lesson plans. This phase aimed to identify the gap between current instructional practices and the potential for integrating programming activities to enhance reasoning abilities.

The *Design* phase focused on developing the conceptual framework for the PjBL module, determining learning objectives aligned with reasoning skill indicators, and planning the experimental procedure. Learning topics were structured around 2D game development projects, including pattern-based drag-and-drop games and a jumping game, which were selected for their potential to engage students in systematic planning and logical sequencing .

The *Development* phase involved the creation and refinement of learning materials, including the PjBL-Scratch module and research instruments. The reasoning ability test (pretest-posttest) was developed based on indicators adapted from Raven's Progressive Matrices , focusing on pattern recognition, logical sequencing, and systematic thinking. All instruments underwent expert validation before pilot testing.

The *Experiment* phase consisted of the five-meeting treatment period. Prior to the intervention, students completed the pretest to establish baseline reasoning ability. During the treatment, students worked in groups of six on game development projects using Scratch, following the PjBL syntax: (1) starting with essential questions, (2) designing project plans, (3) creating schedules, (4) monitoring project progress, (5) assessing outcomes, and (6) evaluating the learning experience .

The *Evaluation* phase concluded the research with posttest administration to measure reasoning ability improvement and distribution of the System Usability Scale (SUS) questionnaire to gather student perceptions of the learning method's practicality.

2.3 Reasoning Ability Test and Learning Module

The primary instrument for measuring reasoning ability was a 10-item multiple-choice test developed by the researchers. The test items were constructed based on indicators adapted from Raven's Coloured Progressive Matrices (CPM) (Muniz et al., 2016), which is widely used to measure logical reasoning and pattern recognition in children. Each item presented a visual pattern matrix requiring students to identify the missing element, thereby assessing their capacity for systematic thinking and relational reasoning. The test was administered twice: as a pretest (before treatment) and as a posttest (after treatment), using equivalent items to ensure comparability of scores.

The learning module was developed to integrate Scratch visual programming with the PjBL model. The module consisted of five sessions aligned with the treatment meetings, covering: (1) introduction to Scratch interface and block-based programming concepts, (2) creating simple animations, (3) developing drag-and-drop games based on number patterns, (4) developing drag-and-drop games based on shape patterns, and (5) creating a jumping game. Each session followed the PjBL syntax and included step-by-step instructions, visual aids, and reflection prompts designed to stimulate reasoning processes.

2.4 System Usability Scale Questionnaire and Project Assessment Rubric

To assess the practicality of the learning method and module from the students' perspective, the System Usability Scale (SUS) was employed. The SUS is a reliable, ten-item Likert-scale questionnaire (ranging from 1 = strongly disagree to 5 = strongly agree) that provides a global measure of system usability (Brooke, 1996). The SUS was selected due to its robustness with small sample sizes and its widespread use in educational technology research. The questionnaire was administered after the posttest, and responses were scored according to the standard SUS scoring procedure, yielding a score between 0 and 100.

To evaluate the quality of student-developed Scratch projects, an analytical rubric was developed. The rubric assessed four dimensions: (1) functionality (whether the game works as intended), (2) logical structure (appropriate use of sequences, loops, and conditionals), (3) creativity (originality in design and implementation), and (4) alignment with project objectives. Each dimension was scored on a scale of 1-4, with descriptors provided for each level.

3 Results and Discussion

3.1 Validity and Practicality of the Developed Module

3.1.1 Expert validation results

The learning module, which integrates Scratch visual programming with the Project-Based Learning (PjBL) model, underwent content validity testing by two expert validators: a media expert and a material expert. This validation process employed Aiken's V coefficient to ensure the module's appropriateness and coherence prior to classroom implementation . Expert evaluation was essential to guarantee that the media aspects, instructional design, and content accuracy met the academic standards required for experimental research .

The results of the expert validation, summarized in Table 2, demonstrated that the module possessed a high level of validity and reliability. The media expert's assessment, spanning 18 items, yielded an average Aiken's V value of 0.70, confirming content validity for the media aspect based on the accepted threshold (V ≥ 0.70) . The reliability test for the media aspect showed a Cronbach's Alpha value of 0.992, substantially exceeding the minimum threshold of 0.60 , indicating excellent internal consistency.

Table 2: Summary of key module validation results

Aspect of Validation	Instrument Used	Number of Items	Average Aiken's V	Cronbach's Alpha	Conclusion
Media Expert	Aiken's V	18	0.70	0.992	Valid & Reliable
Material Expert	Aiken's V	12	0.76	0.944	Valid & Reliable

Similarly, the material expert's assessment, covering 12 items related to instructional design and content accuracy, resulted in an average Aiken's V value of 0.76, confirming content validity for the material aspect. The reliability of the material aspect was also confirmed with a Cronbach's Alpha value of 0.944, well above the acceptable threshold. These results indicate that both the media and material aspects of the module were highly reliable and valid for use in the experiment .

3.1.2 Usability and project performance results

Beyond expert validation, the practicality of the developed module and the PjBL method was assessed through the SUS questionnaire administered to the 30 student participants. The usability evaluation aimed to determine the extent to which the module and method were acceptable and user-friendly for fifth-grade students in practicing reasoning skills .

The results demonstrated high acceptance by the students, indicating that the intervention was practical and effective in its implementation. Both the learning method questionnaire and the learning module questionnaire yielded identical average SUS scores of 86.8 (SD = 5.2). According to the SUS score interpretation guidelines , this value falls within the ”Acceptable” range (scores above 70) and corresponds to a ”Grade A” rating. This high usability score suggests that the integration of the PjBL model and the Scratch module was perceived as easy to use and comprehensible by the primary school students, overcoming common barriers related to complexity often found in technology-based interventions in early education .

Furthermore, the student groups' ability to successfully implement the learned concepts was measured by evaluating their final project performance. The analysis of student project group scores indicated a highly successful outcome, with an average score of 95 out of 100 (SD = 3.8), placing the project results in the ”Very Good” criteria. The projects included the creation of a drag-and-drop game and a jumping game, both of which required systematic planning and logical sequencing. Figure 2 illustrates students' activities while using Scratch for developing game projects.

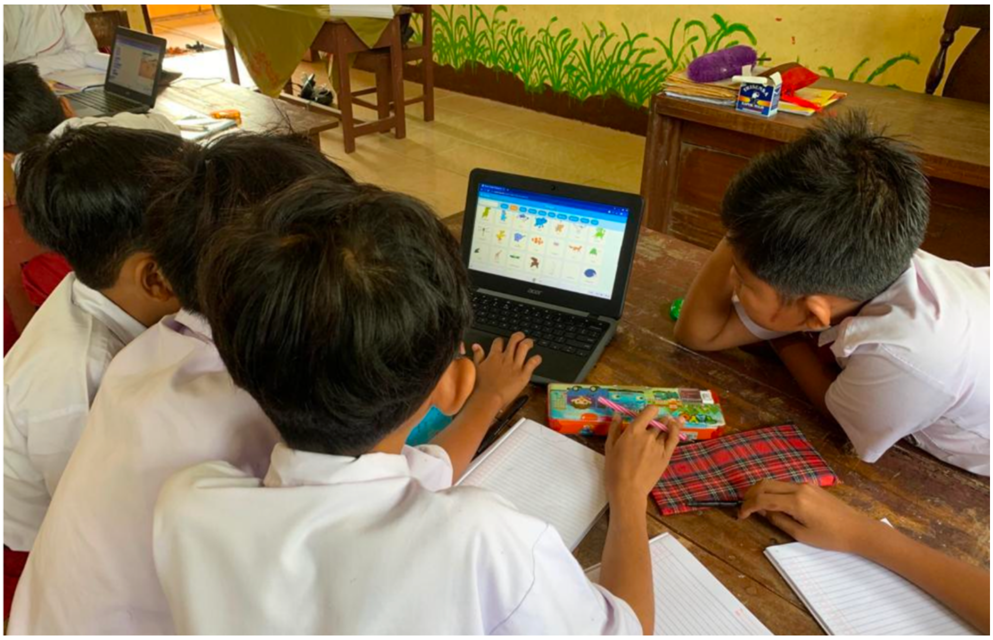

Figure 2: Students’ activities during game development using scratch

3.1.3 Reasoning test instrument validity and reliability

The objective measure of reasoning ability was administered using a pretest and posttest, originally consisting of 10 multiple-choice questions adapted from Raven's Coloured Progressive Matrices . Before analyzing the experimental data, the quality of the test items was confirmed through validity and reliability tests using SPSS with 30 participants (N=30). The validity criterion established that an item was valid if its calculated correlation coefficient (*r*_hitung) exceeded the critical *r*-table value of 0.361 (df = 28, α = 0.05) .

The item validation results, presented in Table 3, showed that 8 out of 10 questions were declared valid, while questions 3 and 5 were determined to be invalid (*r*_hitung < 0.361). Consequently, only the 8 valid items were used for the final analysis of student scores. Following item selection, the reliability of the 8 valid questions was assessed. The final reliability test yielded a Cronbach's Alpha of 0.634, which exceeds the minimum threshold of 0.60 for educational research . Therefore, the reasoning test instrument was deemed reliable and appropriate for measuring the difference in students' reasoning ability between pretest and posttest.

Table 3: *Summary of reasoning test item validity results*

Item Number	*r*_hitung	*r*_tabel (α=0.05)	Validity Status
1	0.412	0.361	Valid
2	0.398	0.361	Valid
3	0.287	0.361	Invalid
4	0.445	0.361	Valid
5	0.312	0.361	Invalid
6	0.421	0.361	Valid
7	0.467	0.361	Valid
8	0.384	0.361	Valid
9	0.403	0.361	Valid
10	0.376	0.361	Valid

3.2 Quantitative Hypothesis Testing Results

3.2.1 Descriptive statistics of reasoning ability

The descriptive analysis was conducted to illustrate the change in students' reasoning ability before and after the intervention using Scratch Visual Programming integrated with PjBL. The reasoning ability was measured using eight validated multiple-choice questions administered as a pretest (*Y_1*) and a posttest (*Y_2*) to the 30 fifth-grade students in the experimental class. The scores demonstrated a significant improvement after the intervention. The mean score of the pretest was recorded at 61, indicating a moderate initial level of reasoning ability among the participants. After undergoing the five-meeting intervention, the mean score dramatically increased to 86.67 in the posttest. This quantitative increase provides preliminary evidence that the application of Scratch programming through 2D game development projects has a positive effect on enhancing students' reasoning skills.

3.2.2 Paired sample T-Test results

To test the primary hypothesis—that Scratch visual programming significantly improves the reasoning skills of primary school students—the Paired Sample T-Test was performed. This inferential statistical test was employed to examine the significance of the difference between the mean pretest and posttest scores . The formal decision rule dictated that if the significance value (Sig. 2-tailed) was less than 0.05, the null hypothesis (H□) would be rejected, and the alternative hypothesis (H□) would be accepted.

The statistical analysis, summarized in Table 4, revealed a highly significant difference between pretest and posttest scores. The mean difference was -25.667 (95% CI: -32.03 to -19.30), with a t-statistic of -8.19 and degrees of freedom (df) of 29. The critical finding was the Sig. (2-tailed) value of 0.000, which is substantially less than the alpha level of 0.05. Therefore, the null hypothesis (H□) was decisively rejected, and the alternative hypothesis (H□) was accepted. This result confirms that there is a statistically significant difference between the pretest and posttest mean scores, implying that the utilization of Scratch visual programming learning strategies effectively improved the reasoning abilities of fifth-grade students.

Table 4: paired sample t-test results

Pair	Mean	SD	Mean Difference	t	df	Sig. (2-tailed)
Pretest	61.00	12.45	-25.667	-8.19	29	0.000
Posttest	86.67	8.92				

3.3 Discussion and Implications

The primary finding of this experimental study, a significant increase in mean reasoning scores from 61.00 (pretest) to 86.67 (posttest), is both statistically significant ($p < 0.001$) and practically meaningful ($d = 1.50$). This result definitively confirms that Visual Scratch Programming learning, implemented through 2D game development projects, effectively improves reasoning skills in primary school students. The large effect size (d = 1.50) observed in this study exceeds those reported in similar interventions, such as the study by Gao et al. (2025) which found moderate effects (d = 0.67) of mind mapping-based scaffolding on elementary students' computational thinking. This finding aligns with a growing body of literature emphasizing the role of programming, particularly block-based visual programming (BBVP) like Scratch (Resnick et al., 2009; Sukirman et al., 2022), in cultivating computational thinking skills that are fundamentally intertwined with reasoning and logical sequencing (Resnick et al., 2009; Wing, 2006).

Reasoning is fundamentally the cognitive process of drawing conclusions or building arguments based on known facts and knowledge (Olsson & Granberg, 2024b). In the context of Scratch programming, students directly apply logical principles by arranging code blocks in specific sequences to achieve desired outcomes, making a sprite move, jump, or respond to user input. This activity inherently trains logical thought and systematic problem-solving, which are key indicators of enhanced reasoning ability (Utami & Ghufron, 2024). The findings corroborate previous research demonstrating that Scratch programming can enhance various cognitive abilities in elementary students. For instance, Utami & Ghufron (2024) argued that Scratch aligns well with children's cognitive development in the concrete operational stage (ages 7-11), as the visual and interactive nature of block-based programming provides concrete representations of abstract logical concepts. Similarly, (Winarsih et al., 2024) reported that Scratch-based activities improved logical reasoning among elementary students in a community service program, although their study lacked rigorous experimental control. The present study extends these findings by providing empirical evidence from a controlled quasi-experimental design with validated instruments.

Furthermore, the improvement in reasoning skills can be explained through the lens of constructionist learning theory, which posits that learners construct knowledge most effectively when they are actively creating meaningful artifacts. In this study, students constructed functional 2D games, thereby ”making their thinking visible” through code. The process of debugging, identifying and fixing errors in code, particularly engages metacognitive processes that require students to reflect on their logical reasoning and revise their mental models. This constructionist approach is further amplified by the integration of PjBL as the instructional strategy. PjBL requires students to engage in complex, meaningful projects, which in this study involved creating functional 2D games such as drag-and-drop games and a jumping game. This finding aligns with research by (Ming, Fuad, Lee, Roziah, et al., 2025), who emphasized that constructionist approaches to coding education, where students create personally meaningful projects, enhance both engagement and learning outcomes.

Creating a game compels students to decompose a complex task (the game mechanics) into smaller, sequential, and logical coding steps—a process fundamental to computational thinking. For instance, designing the drag-and-drop game required students to identify underlying patterns (e.g., number patterns, shape sequences) and translate those patterns into a functional game environment by correctly linking backgrounds (questions) to sprites (answers). This process demands what Olsson & Granberg (2024) term ”creative mathematical reasoning,” where students must construct novel solutions rather than merely recall procedures. Similarly, the jumping game required students to compose block codes so that a sprite moves upward upon key press, incorporating conditional logic (”if key pressed then change y-coordinate") and iterative loops. These programming constructs mirror logical structures found in formal reasoning tasks, such as if-then statements in logical deduction (Gao et al., 2025). The high average group project score of 95 (Very Good criteria) further validates that PjBL provided a meaningful and challenging context that successfully allowed students to apply their burgeoning reasoning and programming skills. The collaborative nature of group work in PjBL also likely contributed to the outcomes. As (Siswandi et al., 2024) noted, PjBL with Scratch coding promotes collaborative problem-solving and peer learning, enabling students to articulate their reasoning to group members and receive feedback. This social construction of knowledge may have enhanced individual reasoning development through what Vygotsky (1978) termed the "zone of proximal development," where students accomplish tasks with peer support that they could not achieve independently.

The feasibility of replicating this intervention in primary education is strongly supported by the high usability scores. The SUS results showed an average score of 86.8 for both the learning method and the learning module, categorizing the resources as "Acceptable" (Grade A) according to established benchmarks. This high acceptability rating suggests that Scratch, combined with the PjBL module, overcomes common barriers related to complexity often found in technology-based interventions in early education, such as syntax errors in text-based programming or overly abstract instructional materials. The usability result is crucial because it demonstrates that both the module and the PjBL method are practical for teachers to implement and easy for students to understand, thereby facilitating the adoption of this innovative teaching approach in computer extracurricular activities. This finding addresses a common concern in educational technology research: that effective interventions often fail to transfer to classroom practice due to poor usability or excessive implementation demands.

The implication of this study is significant for computer education in primary schools: Visual Scratch programming, when applied through a practical, project-based strategy like 2D game development, is a highly effective, tested, and usable medium to train and improve the reasoning ability of elementary students. This approach shifts computer education beyond merely basic operational tasks (such as turning the computer on and off or typing practice) and successfully engages students in higher-order thinking skills required for the 21st century, including critical thinking, problem-solving, and creativity. Moreover, the integration of reasoning skill development with programming aligns with broader educational goals articulated in frameworks such as the Partnership for 21st Century Learning (P21), which identifies critical thinking and problem-solving as essential student outcomes. By embedding reasoning practice within engaging game development projects, this intervention demonstrates how technology education can simultaneously address multiple learning objectives while maintaining high levels of student engagement and usability. The module's expert validation (Aiken's V = 0.70-0.76) and high reliability (α = 0.944-0.992) further ensure that this approach rests on a solid foundation of instructional design principles, making it a viable model for broader implementation in similar educational contexts.

4 Conclusion

This study aimed to investigate the effect of Scratch visual programming, implemented through 2D game development projects with the PjBL model, on the reasoning skills of fifth-grade elementary students. The findings provide compelling empirical evidence that this instructional approach significantly enhances students' reasoning abilities, as demonstrated by the substantial increase in mean scores from pretest (61.00) to posttest (86.67), with a large effect size ($d = 1.50$) and statistical significance ($p < 0.001$). The integration of Scratch programming with PjBL created a learning environment where students actively constructed knowledge by creating meaningful artifacts—functional 2D games—thereby engaging in systematic planning, logical sequencing, and collaborative problem-solving. The high usability scores (SUS = 86.8) and excellent project performance (M = 95) further confirm that this approach is not only effective but also practical and engaging for young learners. Expert validation of the learning module (Aiken's V = 0.70-0.76) and the reasoning test instrument (α = 0.634) ensures the reliability and validity of the research outcomes. These findings have significant implications for elementary computer education, shifting the focus from basic operational tasks to higher-order thinking skills development through creative, project-based programming activities.

Despite the promising results, this study has several limitations that should be acknowledged. First, the one-group pretest-posttest design, while appropriate for exploratory research, lacks a control group, which limits the ability to attribute improvements solely to the intervention. Factors such as maturation, history, or testing effects may have contributed to the observed gains. Second, the study involved a relatively small sample size (N = 30) from a single school in Wonogiri, which restricts the generalizability of findings to other grade levels, school contexts, or demographic populations. Third, the intervention was limited to basic 2D game development (drag-and-drop and jumping games), and the learning activities within the module were relatively limited in scope and variety. Fourth, the study measured reasoning ability immediately after the intervention, leaving unanswered questions about the long-term retention of these cognitive gains. Fifth, potential moderating variables such as prior programming experience, gender differences, or learning styles were not examined, limiting understanding of for whom and under what conditions this intervention is most effective.

Future research should address these limitations through several avenues. First, employing stronger experimental designs with control groups would strengthen causal inferences and better isolate the intervention's effects. Second, replicating the study with larger and more diverse participant populations across multiple schools would enhance the generalizability of findings. Third, expanding the intervention to include more complex game genres and diverse programming challenges could provide richer opportunities for reasoning development. Fourth, developing more comprehensive and varied activity models within the module would maximize student engagement and reasoning practice. Fifth, conducting longitudinal studies with delayed posttests would assess the sustainability of reasoning gains over time. Sixth, investigating individual difference variables (e.g., gender, prior experience, cognitive styles) would help identify moderators of intervention effectiveness. Finally, exploring the integration of Scratch programming with other subject areas (e.g., mathematics, science, language arts) could reveal cross-curricular applications and broader educational benefits. By addressing these directions, future research can build upon the foundation established in this study to further advance the understanding of how visual programming and project-based learning can foster essential 21st-century skills in elementary education.

References

Apriyanto, A., & Sukirman. (2024). Utilization of visual programming scratch to improve problem-solving skills. *AIP Conference Proceedings*, *2926*(1), 020067.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4–7.

Dias Aziz Sukirman, & Pramudita, A. A. (2022). Utaminingsih,"Block-Based Visual Programming as a Tool for Learning the Concepts of Programming for Novices,." *Int. J. Inf. Educ. Technol*, *12*(5), 365–371. https://doi.org/10.18178/ijiet.2022.12.5.1628

Gao, X., Yang, Y., Du, Y., & Sun, D. (2025). Effect of Mind Mapping-Based Scaffolding on Elementary Students' Computational Thinking in Block-Based Programming. *Journal of Educational Computing Research*, *63*(1), 236–271. https://doi.org/10.1177/07356331241303067

Katoningsih, S., & Sunaryo, I. (2020). Programme for International Student Assesment (PISA) as reading literacy standard: Critical thinking skill is priority. *Education, Sustainability And Society*, *3*(1), 08–10.

Koray, A., & Bilgin, E. (2023). The Effect of Block Coding (Scratch) Activities Integrated into the 5E Learning Model in Science Teaching on Students' Computational Thinking Skills and Programming Self-Efficacy. *Science Insights Education Frontiers*, *18*(1), 2825–2845.

Maggin, D. M. (2022). Group experimental and quasi-experimental research designs for special education intervention validation. In *Handbook of Special Education Research, Volume I* (pp. 101–114). Routledge.

Ming, G. K., Fuad, D. R. S. M., Lee, H. L. J., Yaacob, Y. R. M., & Kang, T. N. W. (2025).

Creative Learning Unplugged: A Constructionist Approach to Coding Education Using Scratch. *International Journal of Research and Innovation in Social Science*, *IX*(IIIS), 556–570. https://doi.org/10.47772/IJRISS.2025.903SEDU0032

Monge Roffarello, A., & Sáenz, J. P. (2025). Supporting teachers, engaging students: A collaborative model for K-12 computing education. *Entertainment Computing*, *54*, 100937. https://doi.org/10.1016/j.entcom.2025.100937

Muniz, M., Gomes, C. M. A., & Pasian, S. R. (2016). Factor structure of Raven's coloured progressive matrices. *Psico-USF*, *21*, 259–272.

Olsson, J., & Granberg, C. (2024). Teacher-student interaction supporting students' creative mathematical reasoning during problem solving using Scratch. *Mathematical Thinking and Learning*, *26*(3), 278–305. https://doi.org/10.1080/10986065.2022.2105567

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67.

Romero, E., & Artal-Sevil, J. S. (2021). Scratch: A tool for learning and teaching. *ICERI2021 Proceedings*, 7844–7853.

Siswandi, R., Mahanani, D., Salahudin, M., Isnaini, I., Budiarsani, N. L., & Hafsah, H. (2024). IMPLEMENTATION OF PROJECT BASED LEARNING MODEL BASED ON CODING SCRATCH PRODUCTS TO INCREASE STUDENTS' UNDERSTANDING OF PRESERVING NATIONAL CULTURE. *Abjadia : International Journal of Education*, *9*(2), 301–309. https://doi.org/10.18860/abj.v9i2.27190

Sri Utami. (2024). A Critical Appraisal: Elementary School Coding Education Effectiveness with Scratch in Alignment with Cognitive Development, Analyzed through Piagetian Lens. *Journal of Electrical Systems*, *20*(5s), 1180–1187. https://doi.org/10.52783/jes.2432

Stamatios, P. (2024). Can preschoolers learn computational thinking and coding skills with ScratchJr? A systematic literature review. *International Journal of Educational Reform*, *33*(1), 28–61.

Sukirman, Pramudita, D. A., Afiyanto, A., & Utaminingsih. (2022). Block-Based Visual Programming as a Tool for Learning the Concepts of Programming for Novices. *International Journal of Information and Education Technology*, *12*(5), 365–371. https://doi.org/10.18178/ijiet.2022.12.5.1628

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, *120*, 64–74.

Utami, S., & Ghufron, A. (2024). A critical Appraisal: Elementary School Coding Education Effectiveness with scratch in alignment with Cognitive Development, analyzed through Piagetian Lens. *Journal of Electrical Systems*, *20*(5s), 1180–1187.

Winarsih, N. A. S., Rohman, M. S., Saraswati, G. W., & Mulyanto, E. (2024). Penalaran Logika menggunakan Scratch pada SD Negeri Pendrikan Lor 03. *Abdimasku: Jurnal Pengabdian Masyarakat*, *7*(2), 684–689.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Zacharis, G. K. (2025). Fostering Early Childhood Science Education Through Visual Programming: A Case Study with Pre-service Teachers Using ScratchJr. In *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 69–84). Springer.