

RESEARCH ARTICLE

A Comparative Study of Game-Based Learning and Collaborative Learning in Computer Programming: Impact on Motivation to Learn, Programming Skills, and Learning Achievement

Kannika Daungcharone¹  and Krittawaya Thongkoo^{1,*} 

¹ College of Arts, Media and Technology, Chiang Mai University, Chiang Mai 50200, Thailand

*Corresponding author: Krittawaya Thongkoo, krittawaya.t@cmu.ac.th

Article Information

Article History:

Received: 22 December 2024

Revised: 8 August 2025

Accepted: 24 August 2025

Published: 18 June 2026

Keywords:

Collaborative Learning

Computer Programming Skills

Game-Based Learning

Higher Education

Abstract

Learning computer programming is often perceived as difficult due to abstract concepts, complex syntax, and the challenges associated with algorithm design and debugging. With the rapid growth of technology and digital learning, new instructional approaches have emerged to address these challenges. Game-based learning and collaborative learning are two widely adopted strategies in programming education. This study investigates their impact on students' motivation, programming skills, and learning outcomes. Participants were 198 first-year university students, divided into two groups: a game-based learning group (GBL, $n = 83$) and a collaborative learning group (COL, $n = 115$). Both groups completed a pre-test to assess prior knowledge, studied PHP programming concepts using different instructional approaches, completed programming assignments, and then completed a post-test and a motivation questionnaire. Results indicate that COL students reported significantly higher motivation in the intrinsic motivation, grade motivation, self-determination, and self-efficacy dimensions than GBL students. Regarding programming skills, COL students performed better in identifying outputs, whereas GBL students demonstrated stronger performance in identifying inputs and processes. No significant difference was found in overall learning achievement between the two groups. These findings suggest that collaborative learning enhances engagement and comprehension through teamwork and peer interaction, while game-based learning provides an interactive and enjoyable means of developing technical skills. Consequently, educators should carefully select pedagogical approaches and tools that best suit their instructional context and learners' needs.

1. Introduction

Programming is a crucial skill for many modern careers (Song et al., 2021; Scherer et al., 2020). Developing this skill enhances problem-solving ability, logical thinking, and creativity, which are essential competencies for professional success (Daungcharone et al., 2024). However, learning programming remains challenging because of abstract concepts, complex syntax, algorithm design, and debugging processes. These challenges often lead students to perceive programming as a difficult subject (van Niekerk and Webb, 2016; Hainey and Baxter,

2024).

Advancements in technology and digital learning have transformed programming education to better meet learners' needs. Traditionally, programming was taught through lectures and computer-based exercises (Santilli et al., 2025). In the digital era, programming, recognized as an essential 21st-century skill (Lavi et al., 2021; Dilekçi and Karatay, 2023), is increasingly supported by educational technologies that make learning more effective, engaging, and enjoyable (Liang et al., 2024).

Today, learners have access to a variety of learning

resources, such as interactive coding platforms, game-based learning tools, visual programming tools, collaborative coding platforms, virtual reality learning tools, and AI-powered learning assistants. These technologies help create authentic learning experiences that meet diverse learner needs in the digital age (Ibarra-Torres et al., 2024; Gutiérrez Beltrán and Martínez Arias, 2025; Arnedo-Moreno and García-Solórzano, 2025; Molina et al., 2025).

Among these approaches, game-based learning is one of the most prevalent educational formats. Games enhance learners' engagement with learning materials and foster curiosity. Furthermore, these games provide learners with opportunities to develop their skills in situations that reflect real-world contexts (Hainey and Baxter, 2024; Arnedo-Moreno and García-Solórzano, 2025; Al-Jamili et al., 2024). Similarly, collaborative learning (COL) supports the development of programming skills through teamwork, knowledge sharing, and collective problem-solving. When enhanced with online collaboration tools, it fosters logical reasoning, creativity, and adaptability (Liao et al., 2019; Lapitan et al., 2023).

Previous studies have shown that both game-based learning and collaborative learning environments can effectively enhance students' learning motivation and academic achievement. However, there is a noticeable gap in the research regarding the direct comparison of these two approaches, particularly in the context of learning computer programming. Therefore, this study aims to investigate the effects of game-based learning and collaborative learning environments in a computer programming course. The findings are expected to provide insights into the suitability of these instructional approaches for future teaching and learning.

Three research questions have been formulated as follows:

- RQ1: Are there significant differences in programming learning motivation between students engaged in a game-based learning environment and those engaged in a collaborative learning environment?
- RQ2: Are there significant differences in programming skills between students educated in a game-based learning environment and those educated in a collaborative learning environment?
- RQ3: Are there significant differences in programming learning achievement between students engaged in a game-based learning environment and those engaged in a collaborative learning environment?

2. Related Work

This study reviews previous research on game-based learning, collaborative learning, and educational technologies that aim to enhance students' motivation,

achievement, and performance in computer programming.

2.1 Key Skills in Computer Programming

Computer programming is a fundamental skill that underpins many aspects of modern society, with applications extending across diverse industries and disciplines (Song et al., 2021; Scherer et al., 2020; Daungcharone et al., 2024). The literature identifies several key skills that contribute to programming success. Problem-solving is widely regarded as a cornerstone of programming expertise because it enables learners to decompose complex problems into manageable components (van Niekerk and Webb, 2016; Hainey and Baxter, 2024). Logical thinking refers to the ability to analyze, formulate, and implement effective algorithms for program development (Román-González et al., 2017). Computational and mathematical abilities, particularly in algebra, logic, and discrete mathematics, are also fundamental to many programming concepts, especially those related to algorithm design, optimization, and complexity analysis (Hainey and Baxter, 2024; Román-González et al., 2017).

Moreover, programming proficiency requires mastery of several fundamental concepts that form the foundation of programming expertise. Understanding the syntax of a programming language is essential, including the rules governing input, processing, and output operations. Furthermore, control flow mechanisms such as conditionals, loops, and recursion enable programmers to manage program execution effectively (Hainey and Baxter, 2024).

In addition, Joseph's Programmer Competency Matrix emphasizes several competencies that contribute to programming expertise, including problem decomposition, code organization, code readability, defensive coding, and error handling (Joseph, nd).

2.2 Game-Based Learning Enhances Computer Programming Performance

Many researchers emphasize the importance of integrating academic content with gaming elements to enhance educational computer games. Such games engage learners in enjoyable learning environments while improving their understanding of subject content (Arnedo-Moreno and García-Solórzano, 2025). Educational games exist in various forms and can be designed to support different learning objectives and learner characteristics. Therefore, the selection of educational games should consider both the subject matter and learner needs.

Hamari et al. demonstrated that educational games positively influence learning by increasing challenge and skill development, which subsequently enhance engagement and learning outcomes (Hamari et al., 2016). Koorsse et al. highlighted the effectiveness of computer games in teaching programming concepts

(Koorse et al., 2015). Furthermore, Hainey and Baxter investigated the characteristics of games that influence programming learning in higher education (Hainey and Baxter, 2024). Their findings demonstrated the effectiveness of games in teaching fundamental programming concepts, including variable declaration, programming syntax, conditionals, loops, functions, arithmetic operators, logical operators, and program execution sequences.

The use of educational games as digital learning tools can enhance student satisfaction, engagement, and learning effectiveness while providing meaningful learning experiences that promote motivation and problem-solving skills (Arnedo-Moreno and García-Solórzano, 2025; Chen and Huang, 2024). Educational games leverage their motivational characteristics to address educational challenges and support skill development (Hainey and Baxter, 2024; Cheng et al., 2023). By combining instructional content with interactive and enjoyable experiences, educational games provide a more engaging alternative to traditional instructional approaches (Arnedo-Moreno and García-Solórzano, 2025; Chen and Huang, 2024).

2.3 Fostering Programming Performance with Collaborative Learning

Collaborative learning is an effective instructional strategy for enhancing students' skills and competencies in higher education (Muñoz Miguel et al., 2023). Collaborative learning is an educational approach in which students work together to achieve shared learning objectives. Through collaborative engagement in various tasks and activities, learners can reflect on their understanding, exchange ideas, and develop important social and communication skills through meaningful group interactions (Serrano-Cámara et al., 2014).

This approach promotes student interaction, knowledge sharing, and collaborative knowledge construction while reducing limitations associated with time and location. Online collaborative learning offers a flexible framework that can be effectively implemented in contemporary educational environments. Both face-to-face and online learning settings can benefit from collaborative learning by fostering a sense of belonging and increasing student engagement. Previous studies have shown that collaborative learning positively influences academic performance, motivation, and participation (Liao et al., 2019; Serrano-Cámara et al., 2014).

The effectiveness of online collaborative learning can be attributed to its instructional structure and interactive characteristics, which promote active and meaningful learning experiences. Such environments facilitate the development of higher-order thinking skills, including critical thinking and problem-solving, while enhancing motivation and overall learning satisfaction (Daungcharone et al., 2024; Liao et al., 2019). The flexibility and inclusiveness of collaborative learning make

it a valuable instructional approach for improving engagement, learning outcomes, and skill development in modern educational settings.

The literature highlights the potential benefits of both game-based learning and collaborative learning for programming education. Game-based learning enhances motivation, engagement, and problem-solving skills by integrating interactive and enjoyable learning experiences. It also supports conceptual understanding and makes complex topics more accessible to learners. In contrast, collaborative learning promotes teamwork, communication, and critical thinking through shared learning activities and peer interaction. Both approaches can contribute to increased motivation, active participation, and improved learning outcomes. Therefore, this study compares the effectiveness of game-based learning and collaborative learning in enhancing students' programming skills and examines the suitability of these instructional approaches for future teaching and learning contexts.

3. Research Methodology

This study adopted a quasi-experimental design to investigate the effects of game-based learning (GBL) and collaborative learning (COL) on students' motivation, programming skills, and learning achievement. Data were collected from two groups of students who were assigned to different learning environments through a lottery draw. One group participated in a game-based learning environment, whereas the other group learned through a collaborative learning environment. The research methodology is described in the following subsections.

3.1 Participants

This study involved 198 volunteer first-year university students from Chiang Mai University and Maejo University, aged between 18 and 20 years, who were enrolled in information technology, computer science, and related programs requiring foundational programming knowledge for business applications. All participants shared a similar background in PHP programming for web application development. The demographic characteristics of the participants are summarized in Table 1.

Table 1. Participants' demographic characteristics.

Group	Characteristic	Option	Frequency	Percentage (%)
GBL	Age	18	29	14.65
		19	40	20.20
		20	14	7.07
	Gender	Male	48	24.24
		Female	35	17.68
COL	Age	18	42	21.21
		19	55	27.78
		20	18	9.09
	Gender	Male	69	34.85
		Female	46	23.23

Regarding the learning environment, 83 students were assigned to study computer programming through a game-based learning environment (GBL), whereas 115 students were assigned to a collaborative learning environment (COL). Each student had an equal probability of assignment through a lottery draw, thereby ensuring fairness in the assignment process and reducing potential researcher bias.

3.2 Measurement Tools

Four measurement instruments were developed to assess students' learning outcomes in accordance with the objectives of the study. Pre-test and post-test assessments were used to evaluate students' learning achievement and improvements in PHP programming knowledge before and after the intervention. A rubric-based assessment was employed to evaluate practical programming skills through programming assignments. In addition, a motivation questionnaire was administered to assess students' learning motivation. Details of the four instruments are described below.

The pre-test was administered to evaluate students' PHP programming knowledge before instruction, whereas the post-test assessed their knowledge after the learning process. Both tests were developed by lecturers responsible for teaching the introductory PHP programming course. Each test consisted of 20 multiple-choice questions covering the identification of program inputs, outputs, and processes. The content focused on fundamental PHP programming concepts, including sequence structures (executing statements in a linear order), selection structures (using conditional statements to determine actions based on specified criteria), and repetition structures (executing a code block repeatedly).

The KR-20 coefficient was employed to evaluate the reliability of the pre-test and post-test instruments. KR-20 coefficients of 0.76 and 0.78 for the pre-test and post-test, respectively, indicated acceptable reliability.

The third instrument was a rubric-based assessment designed to evaluate students' programming skills through practical assignments. These assignments covered fundamental PHP programming concepts, including algorithm design using sequence, selection, and repetition structures. The first dimension, identifying input, measured students' understanding of the data required by a program. The second dimension, identifying output, evaluated students' ability to determine the expected output produced after program execution. The third dimension, identifying the process, assessed students' competency in constructing PHP expressions and procedures to solve programming problems.

Table 2 presents the adapted rubric scoring system based on Joseph's Programmer Competency Matrix (Joseph, nd). The rubric evaluated three dimensions of programming skills, with proficiency levels ranging from 0 to 3. A score of 0 indicated insufficient knowledge or skills and suggested a need for additional instruc-

tion and support. A score of 1 reflected basic proficiency, indicating that students possessed limited understanding and required further practice. A score of 2 represented intermediate proficiency, demonstrating reasonable problem-solving ability and conceptual understanding, although additional practice was still necessary. A score of 3 indicated advanced proficiency, demonstrating the ability to solve problems effectively and a thorough understanding of the subject matter.

The fourth instrument was the motivation questionnaire developed by Glynn et al. (2011). The questionnaire was used to assess students' motivation in learning programming analysis and design. It employed a five-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree) and consisted of 20 items divided equally into four dimensions, with five items per dimension.

Intrinsic motivation (IM) refers to the inherent satisfaction derived from learning programming. Example item: "I enjoyed learning about programming and was interested in the topic through game-based learning/-collaborative learning."

Grade motivation (GM) refers to students' desire to achieve high academic performance through acquired knowledge and skills. Example item: "I believe that using game-based learning/collaborative learning will help me achieve a high score in programming."

Self-determination motivation (SDM) refers to students' commitment and effort in managing their own learning process. Example item: "I fully utilized game-based learning/collaborative learning to learn about programming."

Self-efficacy motivation (SEM) refers to students' beliefs in their capability to learn programming successfully. Example item: "I am confident in using game-based learning/collaborative learning to learn about programming."

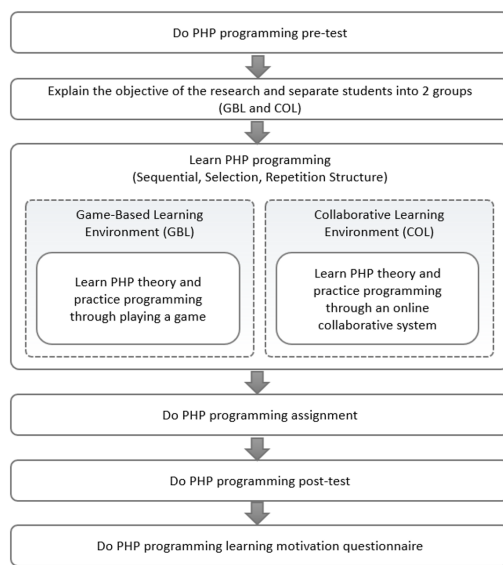
The questionnaire demonstrated high reliability, with a Cronbach's alpha coefficient of 0.91.

3.3 Research Procedures

To achieve the objectives of this study, the effects of game-based learning and collaborative learning on students' motivation, programming skills, and learning achievement were investigated. Fig. 1 illustrates the experimental procedure. The study began with participants completing a 20-minute PHP programming pre-test. Participants volunteered to participate in the study and were subsequently assigned to learning environments through a lottery draw. A total of 83 students were assigned to a game-based learning environment (GBL), whereas 115 students were assigned to a collaborative learning environment (COL).

Table 2. Rubric scoring structure to evaluate PHP programming skills.

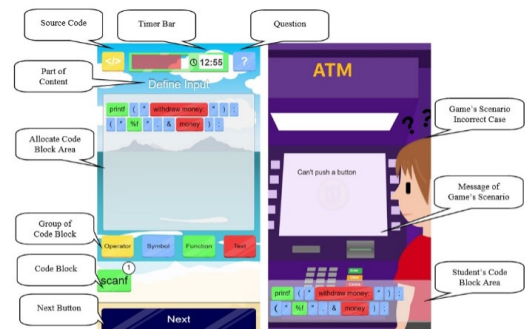
Dimension	Level 0	Level 1	Level 2	Level 3
Input	Unable to define the input.	Can define 50% of the input.	Can define 100% of the input.	Defines 100% of the input systematically and coherently.
Output	Unable to define the output.	Can define 50% of the expected output.	Can define 100% of the expected output.	Outputs 100% of the expected results systematically and coherently.
Process	Unable to create expressions.	Can create 50% of the required expressions.	Creates 100% of the required expressions accurately.	Produces 100% of the required expressions systematically and coherently.

**Figure 1.** Research experiment procedures.

Students in both groups studied PHP programming for five weeks, totaling 20 instructional hours, including 4 hours on sequence structures, 8 hours on selection structures, and 8 hours on repetition structures. Following the instructional period, students completed a 60-minute programming assignment covering fundamental PHP programming concepts. The assignment was designed to evaluate programming skills in three dimensions: identifying input, identifying output, and identifying the programming process. Finally, participants completed a 20-minute PHP programming post-test and a 10-minute learning motivation questionnaire.

To support learning in the game-based learning environment, the CP-Game (Thongkoo et al., 2020) was adopted as the primary learning tool. The CP-Game was designed to assist students in learning, analyzing, designing, and developing computer programs through gameplay. The game aimed to reduce students' perceptions of programming as a difficult subject while encouraging interest, self-directed learning, and knowl-

edge construction. It also helped students understand the stages of computer program execution through everyday scenarios embedded within the game environment. Furthermore, the game assisted students in identifying programming errors and correcting them during the learning process. These interactive scenarios enabled students to learn programming concepts through a step-by-step approach, as illustrated in Fig. 2.

**Figure 2.** The CP-Game overview.

To support collaborative learning, the Logical-Thinking Diagnosis System (LTDS) (Daungcharone et al., 2024) was employed as a learning tool that allowed students to design and develop programs collaboratively with their classmates. The system provided functions for evaluating logical reasoning in programming and delivering feedback on students' strengths and weaknesses. It included a graphical interface that supported the creation and manipulation of flowchart components, such as shapes and connectors, thereby facilitating program planning and debugging before coding.

The LTDS also promoted algorithmic thinking through flowchart-based problem-solving activities. In addition, the system supported online collaborative learning by allowing multiple users to interact with the same flowchart simultaneously in real time. Communication functions, including a chat interface, enabled synchronous discussion and brainstorming among group

members. Consequently, students could learn, practice, and discuss programming concepts collaboratively, learn from mistakes, solve problems with peers, and receive guidance from instructors. An overview of the LTDS is presented in Fig. 3.

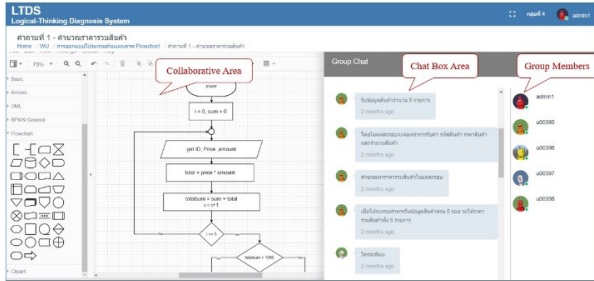


Figure 3. The logical-thinking diagnosis system overview.

4. Experimental Results

This study examines the impact of game-based learning and collaborative learning approaches on students' motivation, programming skills, and learning achievement in computer programming. The results of the experiment are detailed below.

4.1 Result of Learning Motivation

Table 3 presents the results of a one-way MANOVA examining students' motivation to learn computer programming across four dimensions: intrinsic motivation (IM), grade motivation (GM), self-determination motivation (SDM), and self-efficacy motivation (SEM). The analysis compared students in the game-based learning environment with those in the collaborative learning environment. Statistically significant differences were observed between the two learning environments for all four motivational dimensions: intrinsic motivation ($F = 23.880$, $p = 0.006$, $\eta^2 = 0.109$), grade motivation ($F = 27.575$, $p = 0.005$, $\eta^2 = 0.123$), self-determination motivation ($F = 33.532$, $p = 0.007$, $\eta^2 = 0.146$), and self-efficacy motivation ($F = 26.999$, $p = 0.004$, $\eta^2 = 0.121$).

Table 3. Results of one-way MANOVA regarding students' learning motivation.

Dimension	<i>MS</i>	$F_{(1,196)}$	η^2
IM	12.211	23.880*	0.109
GM	14.424	27.575*	0.123
SDM	17.249	33.352*	0.146
SEM	14.144	26.999*	0.121

* $p < 0.05$

Additionally, Table 4 presents the descriptive statistics of students' learning motivation in the two learning environments. The results indicate that students in the collaborative learning environment demonstrated higher levels of intrinsic motivation ($M = 4.06$, $SD = 0.800$), grade motivation ($M = 4.16$, $SD = 0.790$), self-determination motivation ($M = 4.12$, $SD = 0.791$), and self-efficacy motivation ($M = 4.15$, $SD = 0.803$) than students in the game-based learning environment (IM: $M = 3.56$, $SD = 0.576$; GM: $M = 3.62$, $SD = 0.617$; SDM: $M = 3.52$, $SD = 0.598$; SEM: $M = 3.60$, $SD = 0.595$).

Table 4. Descriptive statistics of students' learning motivation by learning environment.

Dimension	Learning Environment	<i>n</i>	<i>M</i>	<i>SD</i>
IM	GBL	83	3.56	0.576
	COL	115	4.06	0.800
GM	GBL	83	3.62	0.617
	COL	115	4.16	0.790
SDM	GBL	83	3.52	0.598
	COL	115	4.12	0.791
SEM	GBL	83	3.60	0.595
	COL	115	4.15	0.803

4.2 Result of Computer Programming Skill

A one-way MANOVA was conducted to compare students' programming skills in three areas: identifying input, identifying output, and identifying program processes. Students in the game-based learning environment were compared with those in the collaborative learning environment. Table 5 presents the results. A statistically significant difference was found in identifying output skills ($F = 23.487$, $p = 0.005$, $\eta^2 = 0.107$). However, no statistically significant differences were observed in identifying input skills ($F = 2.767$, $p = 0.098$, $\eta^2 = 0.014$) or identifying process skills ($F = 0.047$, $p = 0.828$, $\eta^2 = 0.002$) between the two learning environments.

Table 5. One-way MANOVA results for students' programming skills.

Skill	<i>MS</i>	$F_{(1,196)}$	η^2
Input	1.218	2.767	0.014
Output	11.873	23.487*	0.107
Process	0.023	0.047	0.002

* $p < 0.05$

Furthermore, Table 6 presents the descriptive statistics of students' programming skills in the two learning

environments. The findings indicate that students in the collaborative learning environment obtained higher scores in identifying output ($M = 2.56$, $SD = 0.763$) than students in the game-based learning environment ($M = 2.06$, $SD = 0.631$). In contrast, students in the game-based learning environment achieved slightly higher mean scores in identifying input ($M = 2.60$, $SD = 0.680$) and identifying process skills ($M = 2.27$, $SD = 0.842$) than students in the collaborative learning environment ($M = 2.44$, $SD = 0.651$ and $M = 2.25$, $SD = 0.574$, respectively).

Table 6. Descriptive statistics of students' programming skills by learning environment.

Dimension	Learning Environment	n	M	SD
Input	GBL	83	2.60	0.680
	COL	115	2.44	0.651
Output	GBL	83	2.06	0.631
	COL	115	2.56	0.763
Process	GBL	83	2.27	0.842
	COL	115	2.25	0.574

4.3 Result of Learning Achievement

Students' learning achievement was evaluated using a one-way ANCOVA based on PHP programming pre-test and post-test scores. Students in the game-based learning environment were compared with those in the collaborative learning environment. The learning environment was treated as the independent variable, the post-test score as the dependent variable, and the pre-test score as the covariate to control for differences in prior knowledge.

Table 7 shows that the difference in post-test scores between students in the game-based learning environment and those in the collaborative learning environment was not statistically significant ($F = 1.813$, $p = 0.180$, $\eta^2 = 0.009$). Although the difference was not significant, students in the collaborative learning environment achieved a higher mean post-test score ($M = 12.30$, $SD = 3.228$) than students in the game-based learning environment ($M = 11.42$, $SD = 2.741$).

Table 7. One-way ANCOVA results for students' learning achievement.

Learning Environment	n	M	SD	MS	$F_{(1,195)}$	η^2
GBL	83	11.42	2.741	13.781	1.813	0.009
COL	115	12.30	3.228			

5. Discussion

The findings of this study highlight the comparative effectiveness of game-based learning (GBL) and collaborative learning (COL) environments in enhancing students' motivation, programming skills, and learning achievement in computer programming. The results provide valuable insights into the strengths and limitations of these instructional approaches and their implications for programming education.

To answer RQ1, the findings presented in Tables 3 and 4 indicate that students in the COL environment demonstrated higher levels of motivation across all dimensions, including intrinsic motivation, grade motivation, self-determination motivation, and self-efficacy motivation, than students in the GBL environment. These findings suggest that collaborative learning provides a supportive and engaging learning environment that encourages students to participate actively in the learning process. The higher motivation levels observed in the COL environment may be attributed to opportunities for peer interaction, idea exchange, and teamwork.

To answer RQ2, the results reveal mixed findings regarding students' programming skills, as shown in Tables 5 and 6. Students in the COL environment achieved significantly higher scores in identifying program outputs than students in the GBL environment. However, no statistically significant differences were observed between the two groups in identifying program inputs and processes. Although students in the GBL environment obtained slightly higher mean scores in these dimensions, the differences were not statistically significant. These findings suggest that collaborative learning may facilitate conceptual understanding related to program outputs, whereas game-based learning may support procedural thinking and problem-solving activities commonly associated with programming tasks. Such outcomes may be explained by the interactive and immersive characteristics of game-based activities, which often encourage systematic problem-solving and algorithmic reasoning (Hamari et al., 2016; Chen and Huang, 2024). Overall, the findings suggest that both approaches contribute to programming skill development through different instructional mechanisms.

To answer RQ3, the results indicate no statistically significant difference in learning achievement between students in the GBL and COL environments, as shown in Table 7. Although students in the COL environment achieved a slightly higher mean post-test score than students in the GBL environment, the difference was not statistically significant. This finding suggests that both instructional approaches can effectively support student learning, although they may do so through different learning processes (Daungcharone et al., 2024; Hailey and Baxter, 2024; Arnedo-Moreno and García-Solórzano, 2025; Liao et al., 2019). Collaborative learning encourages deeper understanding and re-

flection through peer discussion and group interaction, whereas game-based learning emphasizes active participation and independent problem-solving.

The findings of this study provide practical implications for teachers and curriculum developers seeking to improve programming instruction. The results suggest that both game-based learning and collaborative learning are effective instructional approaches and that combining their strengths may enhance learning outcomes. For example, teachers may employ game-based learning activities to strengthen students' problem-solving and algorithmic reasoning skills while incorporating collaborative learning activities to promote teamwork, communication, and conceptual understanding. Curriculum developers may also consider integrating gamified elements into collaborative tasks or designing project-based learning activities that combine competition and cooperation to sustain student motivation and engagement. Furthermore, teacher professional development programs should emphasize the effective balance between individual and group learning activities, enabling teachers to adapt instructional strategies according to learning objectives, student characteristics, and classroom contexts.

6. Conclusion

This study examined the comparative effects of game-based learning (GBL) and collaborative learning (COL) environments on students' motivation, programming skills, and learning achievement in computer programming. The findings directly address the three research questions and provide insights into how each instructional approach contributes to different aspects of student learning.

For RQ1, students in the COL environment demonstrated higher levels of motivation across all dimensions, including intrinsic motivation, grade motivation, self-determination motivation, and self-efficacy motivation. These findings suggest that collaborative learning can create a supportive learning environment through peer interaction, communication, and shared problem-solving experiences, thereby enhancing student motivation.

For RQ2, the findings revealed partial differences in programming skills between the two learning environments. Students in the COL environment performed better in identifying program outputs, whereas students in the GBL environment obtained slightly higher mean scores in identifying program inputs and processes. These results suggest that collaborative learning may support conceptual understanding, while game-based learning may facilitate procedural thinking and problem-solving skills.

For RQ3, no statistically significant difference in learning achievement was observed between students in the two learning environments. Although students in

the COL environment achieved slightly higher post-test scores, both instructional approaches were effective in supporting learning achievement. This finding suggests that game-based learning and collaborative learning can improve student performance through different instructional mechanisms.

One limitation of this study is that the participants were drawn from a single institution, which may limit the generalizability of the findings. In addition, the intervention was conducted within a single programming course and over a limited instructional period.

Overall, the results indicate that both GBL and COL are effective yet distinct pedagogical approaches for programming education. Collaborative learning appears to be particularly beneficial for enhancing student motivation and conceptual understanding, whereas game-based learning may provide advantages in promoting engagement and problem-solving skills. Future research should investigate hybrid instructional models that integrate the strengths of both approaches and explore how emerging technologies, including AI-assisted educational games and collaborative learning platforms, can further support programming education.

Acknowledgments

This work is an extension of the study by the National Research Council of Thailand (NRCT), RGNS 63-083.

CRedit Authorship Contribution Statement

Kannika Daungcharone: Conceptualization, Methodology, Software, Formal analysis, Validation, Investigation, Data Curation, Visualization, Funding acquisition, Writing – Original Draft. **Krittawaya Thongkoo:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data Curation, Writing – Review Editing.

Declaration of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Al-Jamili, O., Aziz, M., Mohammed, F., Almogahed, A., and Alawadhi, A. (2024). Evaluating the efficacy of computer games-based learning intervention in enhancing english speaking proficiency. *Heliyon*, 10(16):e36440. DOI: 10.1016/j.heliyon.2024.e36440.

- Arnedo-Moreno, J. and García-Solórzano, D. (2025). Programming fun(damentals): Using commercial video games to teach basic coding to adult learners. *Entertainment Computing*, 52:100850. DOI: 10.1016/j.entcom.2024.100850.
- Chen, C.-H. and Huang, K. (2024). The interplay of rewards and competition in digital game-based learning: Effects on intrinsic motivation, game performance and behaviors, and computational thinking. *Learning and Instruction*, 94:101995. DOI: 10.1016/j.learninstruc.2024.101995.
- Cheng, Y.-P., Lai, C.-F., Chen, Y.-T., Wang, W.-S., Huang, Y.-M., and Wu, T.-T. (2023). Enhancing student's computational thinking skills with student-generated questions strategy in a game-based learning platform. *Computers & Education*, 200:104794. DOI: 10.1016/j.compedu.2023.104794.
- Daungcharone, K., Thongkoo, K., and Panjaburee, P. (2024). Integrating thinking process model technique with online collaborative learning to promote programming logical thinking. *International Journal of Information and Education Technology*, 14(3):501–509. DOI: 10.18178/ijiet.2024.14.3.2071.
- Dilekçi, A. and Karatay, H. (2023). The effects of the 21st century skills curriculum on the development of students' creative thinking skills. *Thinking Skills and Creativity*, 47:101229. DOI: 10.1016/j.tsc.2022.101229.
- Glynn, S. M., Brickman, P., Armstrong, N., and Taa-soobshirazi, G. (2011). Science motivation questionnaire ii: Validation with science majors and non-science majors. *Journal of Research in Science Teaching*, 48(10):1159–1176. DOI: 10.1002/tea.20442.
- Gutiérrez Beltrán, E. J. and Martínez Arias, J. C. (2025). Mi superpoder es la programación: A tool for teaching programming to children and youth. *Science of Computer Programming*, 240:103198. DOI: 10.1016/j.scico.2024.103198.
- Hainey, T. and Baxter, G. (2024). A serious game for programming in higher education. *Computers & Education: X Reality*, 4:100061. DOI: 10.1016/j.cexr.2024.100061.
- Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., and Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior*, 54:170–179. DOI: 10.1016/j.chb.2015.07.045.
- Ibarra-Torres, F., Caiza, G., García, M. V., and Barona-Pico, V. (2024). Use of basic programming tools to foster programming logic in university students with school preparation other than computer science. *Procedia Computer Science*, 237:413–419. DOI: 10.1016/j.procs.2024.05.122.
- Joseph, S. (n.d.). Programmer competency matrix. <https://www.sijinjoseph.com/programmer-competency-matrix/>. Retrieved July 18, 2025.
- Koorsse, M., Cilliers, C., and Calitz, A. (2015). Programming assistance tools to support the learning of *it* programming in south african secondary schools. *Computers & Education*, 82:162–178. DOI: 10.1016/j.compedu.2014.11.020.
- Lapitan, L. D. S., J., Chan, A. L. A., Sabarillo, N. S., Sumalinog, D. A. G., and Diaz, J. M. S. (2023). Design, implementation, and evaluation of an online flipped classroom with collaborative learning model in an undergraduate chemical engineering course. *Education for Chemical Engineers*, 43:58–72. DOI: 10.1016/j.ece.2023.01.007.
- Lavi, R., Tal, M., and Dori, Y. J. (2021). Perceptions of *stem* alumni and students on developing 21st century skills through methods of teaching and learning. *Studies in Educational Evaluation*, 70:101002. DOI: 10.1016/j.stueduc.2021.101002.
- Liang, Y. D., Chen, S., Abeysekera, R., O'Sullivan, H., Bray, J., and Keevill-Savage, I. (2024). Examining the adoption of technology-enhanced learning in universities and its effects on student performance, satisfaction, and motivation. *Computers and Education Open*, 7:100223. DOI: 10.1016/j.caeo.2024.100223.
- Liao, C.-W., Chen, C.-H., and Shih, S.-J. (2019). The interactivity of video and collaboration for learning achievement, intrinsic motivation, cognitive load, and behavior patterns in a digital game-based learning environment. *Computers & Education*, 133:43–55. DOI: 10.1016/j.compedu.2019.01.013.
- Molina, A. I., Bravo, C., Gallardo, J., Lacave, C., and Redondo, M. A. (2025). Awareness support in collaborative programming tools: An evaluation based on programmer's perception and eye tracking. *Journal of Systems and Software*, 220:112276. DOI: 10.1016/j.jss.2024.112276.
- Muñoz Miguel, J. P., Simón de Blas, C., Anguita Rodríguez, F., and García Sipols, A. E. (2023). Collaborative learning in management subjects to university students: A multi-level research to identify group profile, engagement and academic performance. *The International Journal of Management Education*, 21(1):100762. DOI: 10.1016/j.ijme.2022.100762.
- Román-González, M., Pérez-González, J.-C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion va-

- lidity of the computational thinking test. *Computers in Human Behavior*, 72:678–691. DOI: 10.1016/j.chb.2016.08.047.
- Santilli, T., Ceccacci, S., Mengoni, M., and Giaconi, C. (2025). Virtual vs. traditional learning in higher education: A systematic review of comparative studies. *Computers & Education*, 227:105214. DOI: 10.1016/j.compedu.2024.105214.
- Scherer, R., Siddiq, F., and Sánchez Viveros, B. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior*, 109:106349. DOI: 10.1016/j.chb.2020.106349.
- Serrano-Cámara, L. M., Paredes-Velasco, M., Alcover, C.-M., and Velázquez-Iturbide, J. (2014). An evaluation of students' motivation in computer-supported collaborative learning of programming concepts. *Computers in Human Behavior*, 31:499–508. DOI: 10.1016/j.chb.2013.04.030.
- Song, D., Hong, H., and Oh, E. Y. (2021). Applying computational analysis of novice learners' computer programming patterns to reveal self-regulated learning, computational thinking, and learning performance. *Computers in Human Behavior*, 120:106746. DOI: 10.1016/j.chb.2021.106746.
- Thongkoo, K., Panjaburee, P., and Daungcharone, K. (2020). Implementation of mobile game-transformed lecture-based approach to promoting *c* programming languages learning. *International Journal of Mobile Learning and Organisation*, 14(2):236–254. DOI: 10.1504/IJMLO.2020.10026332.
- van Niekerk, J. and Webb, P. (2016). The effectiveness of brain-compatible blended learning material in the teaching of programming logic. *Computers & Education*, 103:16–27. DOI: 10.1016/j.compedu.2016.09.008.