Preprints.org

Article

Not peer-reviewed version

The Impact of Gamification and Serious Games on Computational Thinking and Learning Motivation in Scratch Programming Education

Wan Chong Choi * and Chi In Chang

Posted Date: 26 August 2025

doi: 10.20944/preprints202508.1903.v1

Keywords: gamification; serious games; computational thinking; learning motivation; K-12 programming education; block-based programming

Article

The Impact of Gamification and Serious Games on Computational Thinking and Learning Motivation in Scratch Programming Education

Wan Chong Choi [1,2,*] and Chi In Chang [3]

[1] School of Informatics, Computing and Cyber Systems, Northern Arizona University, USA
[2] Department of Computer Science, Illinois Institute of Technology, USA
[3] Department of Psychology, Golden Gate University, USA
* Correspondence: wc388@nau.edu

Abstract

As technology rapidly evolves, educational strategies that bolster computational thinking and stimulate students' motivation for learning programming are becoming essential. This study explored how integrating a block-based programming language and gamified elements influenced primary school students' computational thinking and learning motivation. A quasi-experimental study was conducted with twenty grade three students who received instruction, incorporating gamification elements. During the course, students developed serious games and animations using mBlock, with rewards given as points via Classdojo. The Programming Computational Thinking Scale (PCTS) was used to assess computational thinking abilities, and the Instructional Materials Motivation Survey (IMMS) was used to measure student motivation. A Programming Achievement Test (PAT) was used to evaluate the students' programming performance. Our study found that integrating gamification with mBlock notably enhanced students' computational thinking abilities, especially their comprehension of computational concepts and practices. This approach also significantly improved students' motivation for learning, particularly across the various dimensions of the ARCS motivation model: attention, confidence, relevance, and satisfaction. Moreover, this study uncovered significant correlations between computational thinking, learning motivation, and programming achievement. Our analysis revealed that dimensions within the PCTS were strongly interconnected, suggesting that improvements in one aspect of computational thinking likely led to advancements in others. Similarly, dimensions within the IMMS also demonstrated strong correlations, indicating the interconnected nature of motivational elements in learning. A notable link was identified between the PCTS and IMMS results, underscoring the correlation between advancements in computational thinking and students' motivational experiences. Furthermore, a significant correlation was found between students' confidence and their programming performance, as measured by the PAT, highlighting the comprehensive connection between computational thinking, motivation, and programming achievement. Overall, the study highlighted the importance of incorporating gamification activities and block-based programming in teaching computational thinking concepts and their impact on students' learning motivation. These findings inform educational practice and curricular design for teaching programming in primary schools, potentially improving students' computational thinking skills and motivation for programming.

Keywords: gamification; serious games; computational thinking; learning motivation; K-12 programming education; block-based programming

1. Introduction

As technology rapidly evolves, the need for programming skills has become increasingly important in education. In recent years, different schools worldwide have included programming

 doi:10.20944/preprints202508.1903.v1

courses for K-12 students, including Macao's primary schools. However, many programming curricula focus on teaching programming syntax without emphasizing the importance of computational thinking [1].

Block-based programming languages, such as Scratch and mBlock, emerged as an approach to teaching programming concepts to beginners. They can be used to study computational thinking because they provide a visual and interactive way for learners to understand programming concepts and algorithms [2–4]. Block structures in these environments enable students to concentrate on comprehending the logical flow and structural framework of programming, thereby alleviating the requirement for rote memorization of code syntax.

While block-based programming languages have made programming more accessible to beginners, learning programming can still be challenging [5]. Many students struggle to understand programming concepts, leading to frustration and disengagement [6]. Programming requires complex thinking, skills in applying knowledge broadly, and understanding across several subjects and ideas [4,7,8]. Furthermore, students may lack the motivation to learn programming if they do not see the relevance or applicability of these skills to their lives. Various scholars claim that traditional teaching methods fall short of addressing the challenges encountered by computer programming students [9]. Therefore, designing engaging and effective learning environments that can support students' motivation and computational thinking is essential.

Gamification has become a possible approach to increasing motivation and engagement in programming education [10]. By incorporating gamified elements such as points, badges, and levels, gamification sought to make learning more engaging and enjoyable for students. Gamification can also provide students with clear goals and feedback, which can help them stay motivated throughout the learning process.

Related literature [11] showed that most gamification studies in programming courses used text-based programming languages at the university level. This indicated a research gap in gamified block-based programming, particularly at the primary school level. We addressed this gap by examining how students creating serious games using the mBlock simultaneously affected computational thinking and learning motivation in primary school education. The four research questions were proposed as follows.

(1) How does the integration of gamification in mBlock learning influence students' computational thinking abilities?

(2) How does incorporating gamification into the mBlock learning environment influence students' learning motivation?

(3) What are the outcomes in terms of student achievement after engaging with mBlock through gamification?

(4) What is the relationship between computational thinking, learning motivation, and programming achievement when students engage with mBlock through gamification?

Our study revealed significant findings. We observed notable enhancements in students' understanding of computational concepts and computational practices after the gamified experiment. However, the improvement in computational perspectives could have been more marked, suggesting that further instructional innovations may be needed. Additionally, the results indicated a substantial uplift in students' learning motivation across all the dimensions of the ARCS motivation model post-intervention, highlighting the effectiveness of gamification in creating an engaging and stimulating learning environment.

Moreover, a notable correlation was identified between computational thinking and learning motivation, underscoring the correlation between advancements in computational thinking and students' motivational experiences. Furthermore, a significant correlation was found between students' confidence and programming performance, highlighting the comprehensive connection between computational thinking, motivation, and programming achievement.

 doi:10.20944/preprints202508.1903.v1

The remainder of the paper is organized as follows: Section 2 presents relevant literature. Section 3 outlines the research methodology, including design, participants, teaching materials, and instruments. Section 4 details the results. Section 5 discusses the findings, and Section 6 summarizes the key findings and implications for future research.

2. Literature Review

2.1. Gamification, Serious Games and Programming Learning

Gamification and serious games have been used as an active educational methodology [12]. Students learn through play from the start of their lives [13]. Games can engage and sustain students' excitement for learning and keep them interested and involved in their learning activities [14,15].

Game-based learning helped bridge the gap between gaming and education. It created a union between the educational component and the gaming concept to make learning fun [15]. Researchers have found that the design elements that went into developing games had cognitive demands on the learner. The learner creates a personal connection with the new knowledge to construct meaningful game designs [16–18].

Moreover, gamification was applied in different subject areas, such as mathematics [19], physics [20], Chinese language [21], and so on. Other literature reviews pointed out that gamification in education was growing in importance, mainly at the university level [22–24].

Gamification was also applied in programming learning. Areed et al. [25] investigated the effectiveness of integrating App Inventor, a programming tool, to enrich university students' programming education and overall learning experiences in a gamified learning environment. Llorent-Vaquero [26] assessed the experience of using Scratch to develop serious games at the university level.

Dwi Maryono et al. [11] studied forty-one papers on applying gamification in programming courses. Most of the studies used text-based programming languages at the university level. This may indicate a research gap in gamified block-based programming, particularly at the primary school level.

We addressed this gap by examining the effects of gamification on computational thinking and learning motivation using mBlock. This study not only differentiated itself from prior research but also aimed to offer insightful recommendations for applying gamification in primary education, thereby making a significant contribution to the academic field.

2.2. Computational Thinking

The concept of “Computational Thinking” was introduced by Wing in 2006 and has gained attention in education. It involves using abstraction, decomposition, and analysis skills to solve problems [27]. Hemmendinger [28] emphasized that the aim of teaching computational thinking should not be to make everyone a computer scientist or software engineer. Instead, the focus should be on teaching individuals to apply computational thinking concepts to solve problems and discover new questions in various disciplines.

Brennan and Resnick have developed a computational thinking framework [4] to facilitate a thorough comprehension of computational thinking, encompassing the following dimensions:

(1) Computational Concepts: This aspect emphasizes learners' need to grasp fundamental programming principles. These include understanding sequences for task organization, parallelism for simultaneous task execution, events for triggering responses, conditions for decision-making, data management for information handling, operators for performing computations, and loops for repetitive action execution.

(2) Computational Practices: This dimension focuses on applying and iterating computational concepts. Learners are encouraged to engage in programming activities, allowing them to refine their debugging and problem-solving skills iteratively.

 doi:10.20944/preprints202508.1903.v1

(3) Computational Perspectives: This involves cultivating a multifaceted view of computation, comprising expression for creative and representational articulation, connection for collaborative and communicative engagement, and questioning for inquisitive exploration and problem comprehension.

Various studies have emphasized that block-based programming tools like Scratch or Code.org could enhance the development of computational thinking skills [29–31]. For instance, Choi et al. [32] report positive links between Code.org use and students' computational thinking, motivation, attitudes, and achievement.

2.3. Learning Motivation

Motivation is a fundamental theoretical concept that sheds light on human actions by prompting individuals to respond to their necessities [33,34]. It involves the initiation, guidance, and persistence of behaviors aimed at objectives, propelling individuals to undertake measures toward completing a goal. Researchers pinpointed motivation as a pivotal element impacting students' learning achievements [35–37].

The high failure and dropout rates in introductory computer programming courses could be attributed to students' lack of motivation to engage [38]. To address this issue, gamification has been proposed as a potential motivational strategy in programming teaching, as it offers interactive, entertaining, and creative experiences [33].

Two gamification approaches have been utilized: game creation and game playing. In the first approach, students developed small games to apply programming concepts [34,39], while in the second approach, students played games to reinforce and practice programming skills and concepts [40,41]. Gamification aims to motivate students to engage in learning activities, reduce the gap between theory and practice, and bridge the divide between abstract concepts and concrete activities. In this study, we focused on students creating simple games and did not consider using existing games for computer programming education.

John M. Keller proposed the ARCS model [42] based on the expectancy-value theory. While these theories had diverse applications, their fundamental premise was that beliefs and expectations influence human behavior. Consequently, motivation was fostered through expectancies, thoughts about the likelihood of success in a given task, and the corresponding subjective value, the level of satisfaction expected from the task's outcome.

3. Methodology

3.1. Research Design

The study lasted seven weeks and adopted a quasi-experimental approach using a single-group pretest-posttest design [43]. This method was selected to evaluate how gamification influences students' computational thinking and motivation to learn. By comparing results before and after the intervention, this setup allowed the researchers to discern possible cause-and-effect relationships and assess the effectiveness of the gamification approach.

The Programming Computational Thinking Scale (PCTS) was used to assess computational thinking, the Instructional Materials Motivation Survey (IMMS) measured learning motivation, and a Programming Achievement Test (PAT) evaluated the students' overall programming achievement. Table 1 shows the details of the research design.

Table 1. Pretest-Posttest Design.

N	Pre-Test	Experiment	Post-Test
20	O_1 O_2	E	O_3 O_4 O_5

E: Indicates the application of gamified programming as the experimental treatment. O_1: Represents the initial evaluation of students' computational thinking through the PCTS pre-test. O_2: Represents the initial evaluation

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 August 2025 doi:10.20944/preprints202508.1903.v1

of students' learning motivation through the IMMS pre-test. O_3: Represents the follow-up evaluation of students' computational thinking through the PCTS post-test after the experiment. O_4: Represents the follow-up evaluation of student's learning motivation through the IMMS post-test after the experiment. O_5: Indicates assessing students' programming achievement through the PAT.

3.2. Participants

The research was carried out at a primary school in Macao, involving 20 students aged 8 to 9 years from grade three. These students came with a foundational background in programming, having been introduced to unplugged coding activities and beginner-friendly programming languages in grade two. Their prior exposure included engaging with tangible programming platforms such as Beebot and Lightbot, designed to introduce young learners to the basics of programming concepts in an interactive and accessible manner. This foundational experience gave the students a preliminary understanding of programming principles, which served as a basis for further exploration and learning in this study.

3.3. Implementation of Experiment

The instructional period lasted seven weeks, with six sessions each week, for a total of forty-two sessions. Each session lasted 40 minutes. The curriculum focused on teaching block-based programming using mBlock and covered topics on computational thinking. Figure 1 showcases a selection of games devised by the students.

During the teaching experiment, the initial phase involved introducing the mBlock environment. Basic block-based programming concepts, such as sequences, parallelism, and events, were introduced to the students. Afterward, other programming concepts, including conditions, data management, operators, and loops, were covered in the curriculum. The students were then tasked with developing serious games and animations utilizing these programming concepts.

Subsequently, the students were tasked to create games encompassing score calculation and increased complexity. Beyond fulfilling the prescribed functions outlined by the instructor, students were encouraged to employ their creativity to further enhance the depth and intricacy of game design during independent work.

Following this creative process, students were allowed to present their individually crafted games to their peers in the classroom. Figure 2 illustrates the graphical user interface of the mBlock.

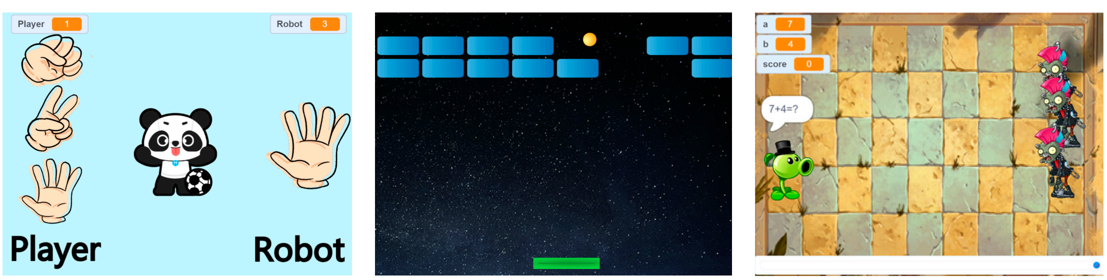


Figure 1. Some serious games created by the students.

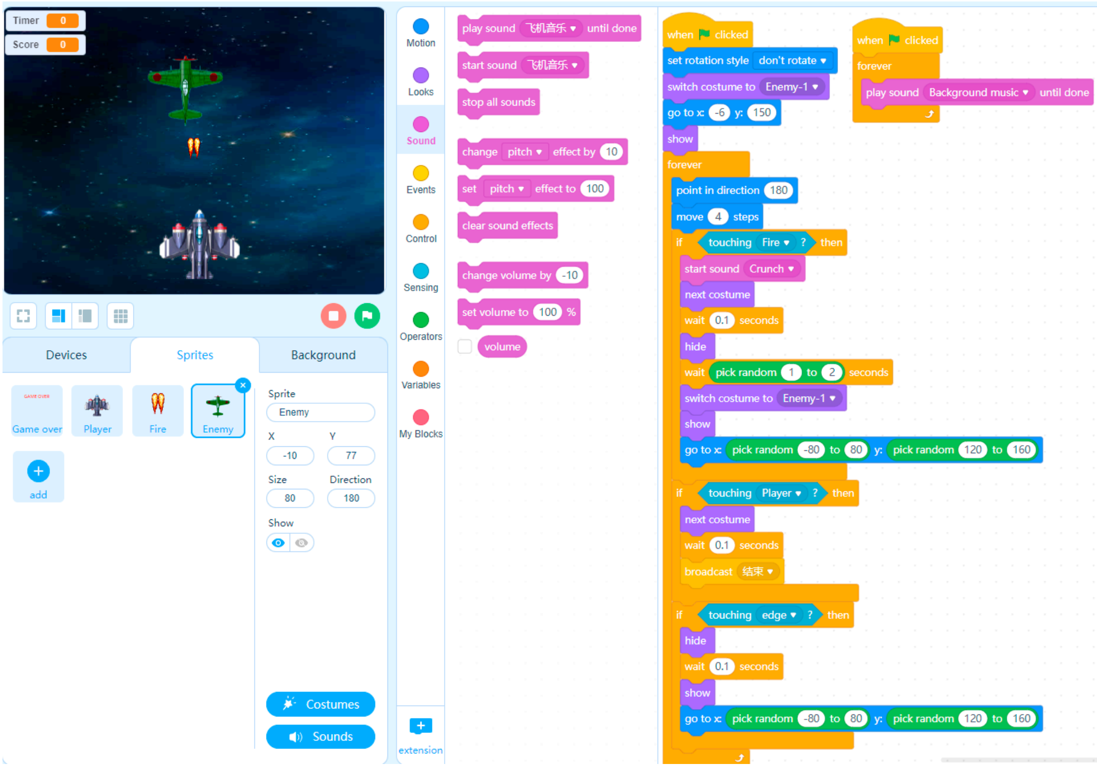


Figure 2. The graphical user interface of the mBlock.

3.4. Integrating Classdojo for Engagement

To enhance engagement in the teaching experiment, students earned points for behaviors like completing coding tasks or presentations, reflected on their Classdojo avatars, as shown in Figure 3.

ClassDojo [44] is a classroom management platform to assist educators in enhancing student behavior and fostering a more positive learning setting. The platform allows teachers to award points to students for specific behaviors or achievements, which are visually represented through personalized avatars. These avatars, which are cartoon characters, along with features like immediate feedback and achievement recognition, aim to motivate students by making them more invested in their learning and aware of their performance.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 August 2025 doi:10.20944/preprints202508.1903.v1

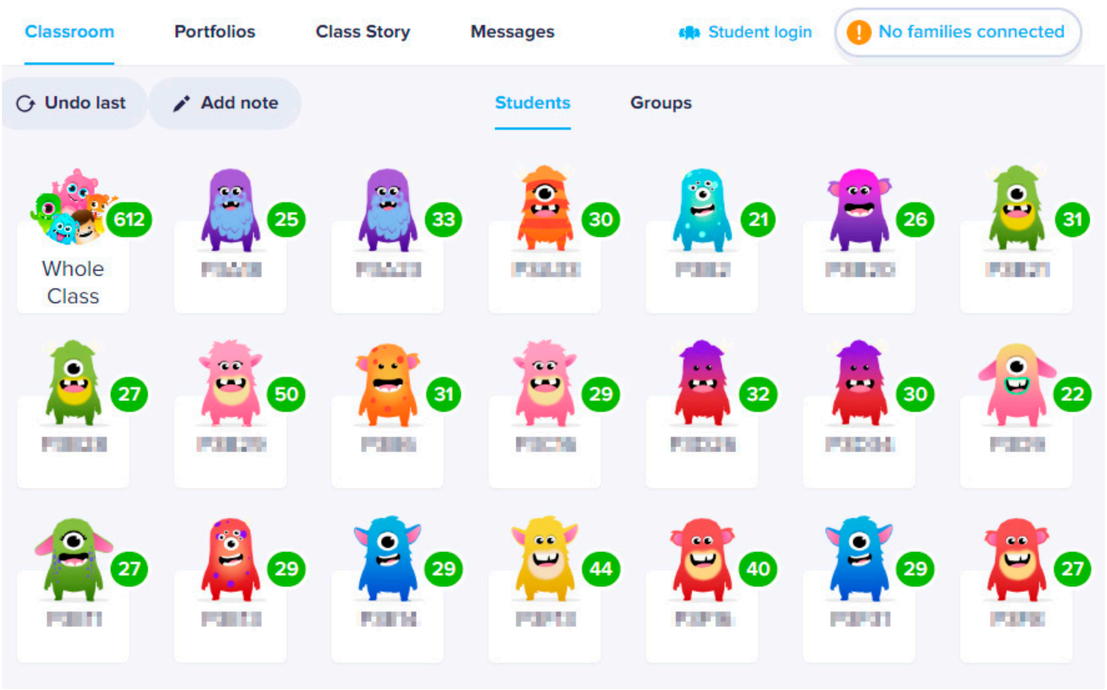


Figure 3. Interface of Classdojo.

3.5. Instruments

3.5.1. Programming Computational Thinking Scale (PCTS)

We incorporated Wu Pei Zhen's PCTS [45], explicitly designed for programming curricula. Adhering to the computational thinking framework from MIT [4] and research by Zhong et al. [46], the PCTS encompasses three dimensions. The scale includes 8 items on Computational Concepts, 8 on Computational Practices, and 5 on Computational Perspectives, summing up to 21 items. The scale's reliability was robust, demonstrated by a Cronbach Alpha of .95.

3.5.2. Instructional Materials Motivation Survey (IMMS)

The study employed Keller's IMMS [47] to assess student motivation in programming learning. The IMMS, comprising 36 items, sets four motivational dimensions: The scale includes 12 items on Attention, 9 items on Relevance, 9 items on Confidence, and 6 items on Satisfaction. The IMMS was known for its high internal consistency in diverse educational settings, with an overall Cronbach Alpha of .95, indicating strong reliability.

3.5.3. Programming Achievement Test (PAT)

In the final session of the instructional period, students were tested using the PAT, which included fifty multiple-choice questions and one practical programming task. It was a regular assessment of students' programming achievement. The teacher assessed the completeness and accuracy of each submission.

3.6. Data Analysis

The data analysis was performed using SPSS. Initially, a normality test confirmed that the data followed a normal distribution. Subsequently, we performed paired samples t-tests to explore any change in students' computational thinking and learning motivation after the experiment. Additionally, we compared the students' overall outcomes using scores from the PAT. Furthermore,

 doi:10.20944/preprints202508.1903.v1

we incorporated a Pearson correlation test to identify the correlation among PCTS, IMMS, and PAT scores.

For a more precise visualization of the results, we used Python libraries, Seaborn and Matplotlib, to create boxplots and heatmaps. The boxplots depicted the distribution of scores and the outcomes of the t-tests. The heatmaps visually represented the correlations, enhancing our understanding of the relationships between these variables.

4. Results

4.1. Analysis of Computational Thinking

The distinction in PCTS scores before and after the experiment was assessed using a paired samples t-test.

Table 2. Paired Samples T-Test Results of PCTS.

Dimension	Test	Mean	N	Sd	t	p
Computational Concepts	Pre-test	3.67	20	0.64	-3.77	0.001
	Post-test	4.31	20	0.56		
Computational Practices	Pre-test	3.66	20	0.62	-2.56	0.019
	Post-test	4.13	20	0.68		
Computational Perspectives	Pre-test	3.75	20	0.72	-0.73	0.474
	Post-test	3.89	20	0.66		
Total	Pre-test	3.68	20	0.62	-2.72	0.014
	Post-test	4.14	20	0.59		

As shown in Table 2, the results highlight significant changes in the PCTS scores across the different dimensions of computational thinking.

Computational Concepts: The pre-test scores had a mean of 3.67 (SD = 0.64), which increased to a post-test mean of 4.31 (SD = 0.56). The paired samples t-test yielded a value of -3.77, with a significant p-value of 0.001 (< 0.05). This indicated a statistically significant improvement in the Computational Concepts dimension following the experiment.

Computational Practices: In this dimension, the pre-test mean was 3.66 (SD = 0.62), which rose to a post-test mean of 4.13 (SD = 0.68). The t-test result was -2.56, with a significant p-value of 0.019 (< 0.05), signifying that the students' Computational Practices significantly improved after the gamified experiment.

Computational Perspectives: The scores showed a less notable change, with a pre-test mean of 3.75 (SD = 0.72) and a post-test mean of 3.89 (SD = 0.66). The t-value was -0.73, with a p-value of 0.474 (> 0.05), which did not suggest a statistically significant difference in the Computational Perspectives.

Overall, the total scores increased from a pre-test mean of 3.68 (SD = 0.62) to a post-test mean of 4.14 (SD = 0.59). The paired samples t-test showed a value of -2.72, with a significant p-value of 0.014 (< 0.05), underscoring a significant overall enhancement in the PCTS scores after the gamified experiment.

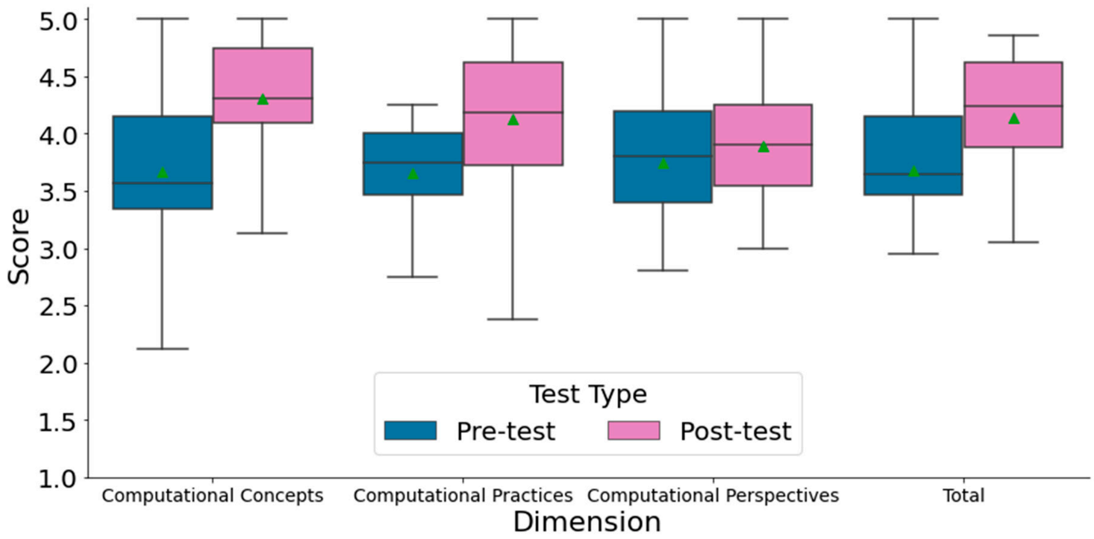


Figure 4. Boxplot for PCTS pre-test and post-test.

Figure 4 presents a boxplot comparison of the PCTS pre-test and post-test scores, clearly representing the statistical results detailed in Table 2. The boxplots elucidated the significant improvement in median scores from pre-test to post-test for the dimensions of Computational Concepts, Computational Practices, and the total score, as indicated by the upward shift in the central mark within the boxes.

The spread of the scores, illustrated by the interquartile range, tightened for Computational Concepts and the total score in the post-test, suggesting a more consistent performance among participants after the intervention.

Although the median improved for Computational Practices, the interquartile range slightly expanded, reflecting a greater diversity in participant scores.

Conversely, the Computational Perspectives dimension displayed a minor median increase with a post-test interquartile range that remained broadly similar to the pre-test, visually reinforcing the statistical findings that the change was insignificant.

This boxplot effectively underscored the overall positive shift in PCTS scores post-experiment, highlighting that the impact on Computational Perspectives was not as pronounced as in the other areas.

In summary, the findings showed that using the mBlock, combined with gamification, has proven to be a successful strategy for enhancing students' comprehension of computational concepts and their ability to apply them in practical programming problems. However, for the Computational Perspectives dimension, the gamification approach utilized in the study might have been less effective in promoting students' perspectives on computational thinking than the other assessed dimensions.

4.2. Analysis of Learning Motivation

The distinction in IMMS scores before and after the experiment was assessed using a paired samples t-test.

Table 3. Paired Samples T-Test Results of IMMS.

Dimension	Test	N	Mean	SD	t	p
Attention	Pre-test	3.72	20	0.39	-3.22	0.005
	Post-test	4.24	20	0.76		

Relevance	Pre-test	3.74	20	0.48	-5.64	0.000
	Post-test	4.46	20	0.45		
Confidence	Pre-test	3.52	20	0.37	-3.72	0.001
	Post-test	4.05	20	0.71		
Satisfaction	Pre-test	3.87	20	0.38	-3.95	0.001
	Post-test	4.47	20	0.66		
Total	Pre-test	3.70	20	0.36	-4.59	0.000
	Post-test	4.28	20	0.60		

As presented in Table 3, the results underscore notable increases in the IMMS scores across the different dimensions of learning motivation.

Attention: The mean pre-test score was 3.72 (SD = 0.39), which increased to a post-test mean of 4.24 (SD = 0.76). The paired samples t-test reported a value of -3.22, with a significant p-value of 0.005 (< 0.05). This marks a statistically significant improvement in the Attention dimension following the experiment.

Relevance: For this dimension, the pre-test mean stood at 3.74 (SD = 0.48) and rose to a post-test mean of 4.46 (SD = 0.45). The t-test outcome was -5.64, with a highly significant p-value of less than 0.001 (< 0.05), indicating a significant enhancement in the students' perception of Relevance after the gamified experiment.

Confidence: The mean score for Confidence moved from a pre-test average of 3.52 (SD = 0.37) to a post-test average of 4.05 (SD = 0.71). The t-value reached -3.72, with a significant p-value of 0.001 (< 0.05), demonstrating a significant increase in student Confidence.

Satisfaction: Pre-test scores for Satisfaction had a mean of 3.87 (SD = 0.38), which improved to a post-test mean of 4.47 (SD = 0.66). The t-test yielded a value of -3.95, with a p-value of 0.001 (< 0.05), reflecting a significant rise in student Satisfaction.

The overall scores showed an upturn from a pre-test mean of 3.70 (SD = 0.36) to a post-test mean of 4.28 (SD = 0.60). The t-test result was -4.59, with a p-value of less than 0.001 (< 0.05), emphasizing a significant general enhancement in IMMS scores after the intervention.

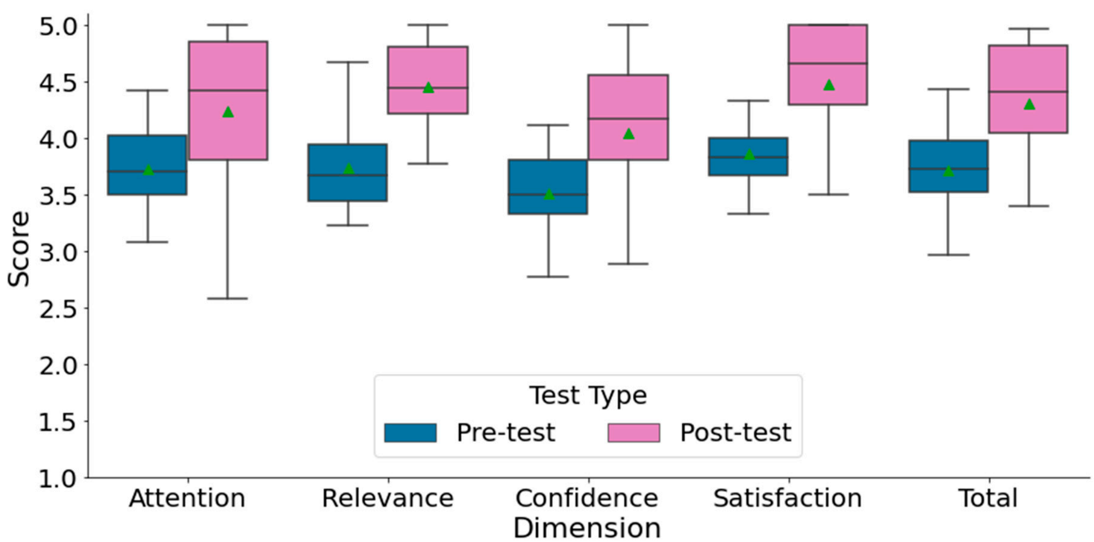


Figure 5. Boxplot for IMMS pre-test and post-test.

Figure 5 presents the boxplot comparison for IMMS pre-test and post-test scores, visually representing the shifts in students' motivation dimensions following the intervention.

 doi:10.20944/preprints202508.1903.v1

The boxplots for Attention show an initial group of scores that suggests a concentrated range during the pre-test phase, with a median just above the scale's mid-point. In the post-test, the median score increased, and there was a broader spread of scores, which visually emphasized the statistical improvement noted in the paired samples t-test results.

For Relevance, the pre-test scores clustered around a median slightly below the post-test median, but the post-test scores not only rose higher but also grouped more closely around the median, indicating a more uniform appraisal of the material's relevance among students after the intervention.

The Confidence dimension displayed a notable upward movement of the median score in the post-test, with the spread of scores extending further than in the pre-test. This suggested that while students generally felt more confident after the intervention, their experiences varied.

Satisfaction scores reflected a similar pattern, with the post-test median visibly higher than the pre-test median and the scores extending further from the median. This suggested that while students were generally more satisfied post-intervention, there was also a more comprehensive range of satisfaction levels.

Overall, the total scores exhibited a significant median increase and a moderate broadening of scores in the post-test. The change in median positions and the range of scores aligned with the significant improvements reported in the t-test results.

These boxplots concisely illustrate the substantial positive shifts in all dimensions of IMMS after the educational intervention, pointing to the success of gamification in boosting student motivation. The visual data demonstrated an increase in median values across all dimensions and variations in the distributions, reflecting the different levels of impact the intervention had on the various aspects of students' motivational experiences.

In summary, these results demonstrate the efficacy of gamification activities in promoting students' interest in learning. In particular, the task of designing a game using the programming concepts learned in class proved effective in mitigating students' negative attitudes toward programming and increasing their recognition of the importance of programming. Classdojo motivated students by fostering a sense of accomplishment and healthy competition with peers, encouraging them to strive for excellence in their learning journey.

4.3. Analysis of Programming Achievement

In the final session of the instructional period, students engaged in a comprehensive assessment consisting of fifty multiple-choice questions and a practical programming exercise. The teacher assessed these tasks, focusing on their completion and accuracy.

The results in Table 4 reveal that most students completed the Sequence of multiple-choice questions, achieving a completion rate of 97%. Furthermore, a significant proportion of students answered correctly to questions related to Events (82%), Data (87.5%), and Operators (80.1%), with completion rates exceeding 80%. However, students exhibited a relatively lower level of proficiency in the areas of Parallelism (73%), Conditions (73%), and Loops (73.9%), probably because these are more abstract concepts. As such, further practice and instruction may be required to facilitate a deeper understanding of these concepts.

Table 4. Result of PAT.

Concept	Total	Mean	Correctness Rate
Sequences	10	9.70	97.0%
Parallelism	10	7.30	73.0%
Events	10	8.20	82.0%
Conditions	20	14.59	73.0%
Data	17	14.88	87.5%
Operators	20	16.01	80.1%

doi:10.20944/preprints202508.1903.v1

Loops	13	9.61	73.9%
Total	100	80.29	80.3%

4.4. Correlation Between Computational Thinking, Learning Motivation, and Programming Achievement

The Pearson correlation test was conducted to explore the relationships among the PCTS, IMMS, and PAT. The results of the Pearson correlation test are presented in Figure 6. The analysis of these results can be found in the following sections.

4.4.1. Significant Correlations within PCTS Dimensions

The dimensions within the PCTS showed strong and significant correlations, indicating that enhancements in one dimension of computational thinking were likely to be accompanied by improvements in others. Notably, the correlation between Computational Concepts and Computational Practices was robust (0.890**), suggesting a close relationship between understanding and applying these concepts in computational practices. Similarly, the overall PCTS score correlated highly with its dimensions, especially with Computational Concepts (0.948**), Computational Practices (0.972**), and Computational Perspectives (0.897**), indicating a cohesive structure of computational thinking assessment.

4.4.2. Significant Correlations Within IMMS Dimensions

Within the IMMS, the dimensions also exhibited strong correlations with each other, reflecting the interconnected nature of motivational aspects in learning.

For instance, the Attention dimension of the IMMS had a very high correlation with the overall IMMS score (0.951**), demonstrating that higher levels of student attention are associated with higher overall motivation towards learning programming. Similarly, the Relevance and Satisfaction dimensions are highly correlated (0.897**). This correlation indicated a strong association between students perceiving learning material as relevant and experiencing increased satisfaction with learning.

4.4.3. Significant Correlations Between PCTS and IMMS Dimensions

The analysis showed significant correlations between the dimensions of PCTS and IMMS, highlighting the interplay between computational thinking skills and motivational factors in programming education.

For example, there was a notable correlation between Computational Concepts and two other IMMS factors: Confidence (0.692**) and Satisfaction (0.854**). These correlations suggest a relationship where students with a deeper understanding of computational concepts also tend to report higher confidence in their programming skills and greater satisfaction with their educational journey. The results emphasized the link between enhanced computational thinking skills and positive motivational outcomes, such as confidence and satisfaction, contributing to a more engaging learning experience.

4.4.4. Overall PCTS and IMMS Correlation

The overall scores of PCTS and IMMS were significantly correlated (0.748**), indicating a notable link between students' computational thinking skills and their motivation toward programming had been observed.

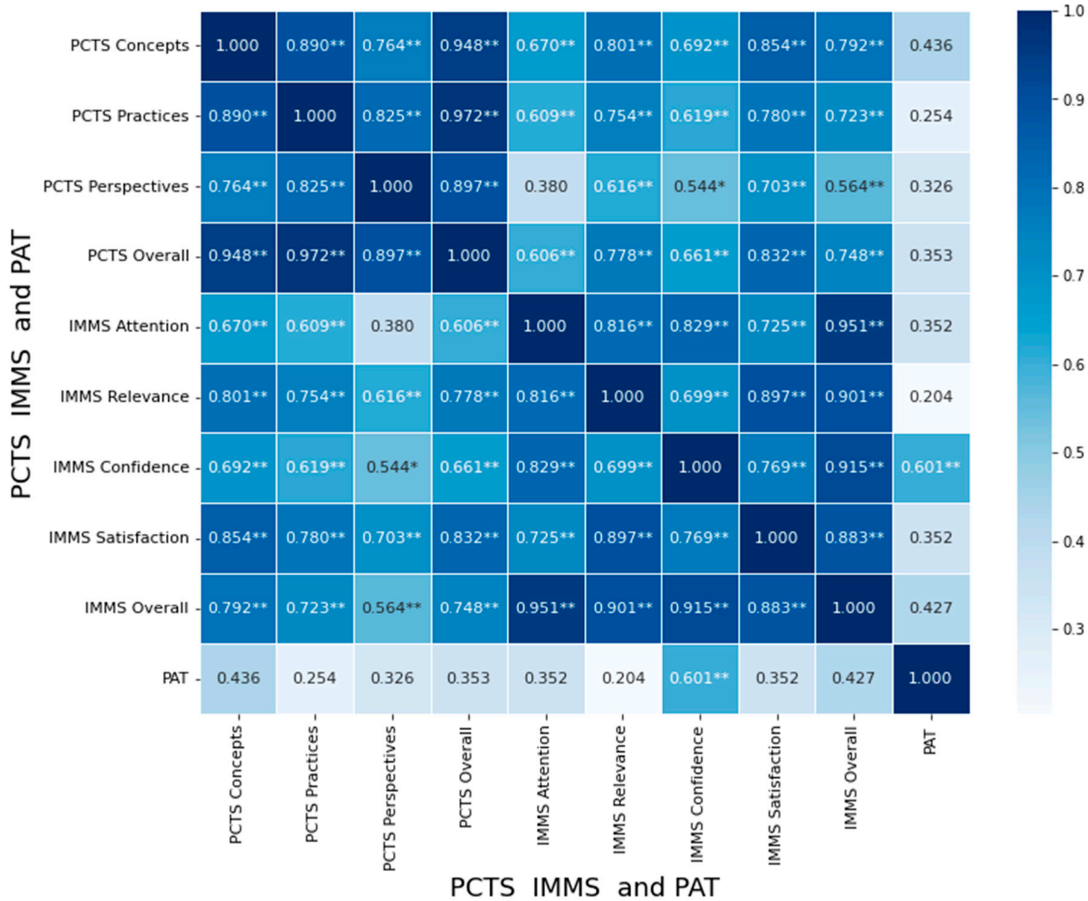
This correlation supported the observation that students with more advanced computational thinking skills also tended to show higher motivation levels to engage with programming material. It reflected a simultaneous relationship where advancements in computational thinking and increased motivation toward programming material were associated, but without implying that one directly caused the increase in the other.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 August 2025 doi:10.20944/preprints202508.1903.v1

4.4.5. Correlation Between PCTS, IMMS, and PAT

The correlations among the constructs of PCTS, IMMS, and PAT illustrated how these aspects related to each other, with the correlation between IMMS Confidence and PAT standing out significantly (0.601**). This relationship underscored the association between students' confidence in their actual performance in programming achievements.

While this finding suggested a strong association, it was necessary to note that other motivational and computational thinking dimensions showed correlations with PAT to varying extents, though not all reached statistical significance. This indicated that confidence was a crucial aspect of performance in programming tasks. However, the direct association between other dimensions of motivation and computational thinking with programming achievement might be more complex and influenced by additional factors, which need deeper investigation in future research.



** denoted that the correlation is significant at the 0.01 level

* denoted that the correlation is significant at the 0.05 level

Figure 6. Pearson correlation heatmap between PCTS, IMMS, and PAT.

5. Discussion

The results are discussed in the following sections, organized according to the research question that guided this research.

5.1. RQ1) How Does the Integration of Gamification in mBlock Learning Influence Students' Computational Thinking Abilities?

The results from the paired samples t-tests on the PCTS indicated a significant enhancement in students' understanding of Computational Concepts and Computational Practices. This improvement demonstrated the positive impact of the integration of gamification in mBlock learning on students' computational thinking abilities. Specifically, it highlighted how gamification, with its engaging and interactive elements, effectively facilitated a deeper understanding of programming logic and structures, making complex ideas more accessible to students.

However, the study also noted that improvements in Computational Perspectives could have been more marked. This suggested that while gamification excelled in teaching the technical and procedural aspects of computational thinking, it might not have fully addressed the broader impacts and creative applications of computing. This observation pointed to the need for further pedagogical strategies to enhance these areas of computational thinking alongside gamified learning, ensuring a more comprehensive development of students' computational thinking skills.

5.2. RQ2) How Does Incorporating Gamification into the mBlock Learning Environment Influence Students' Learning Motivation?

The paired samples t-test results of IMMS showed that integrating gamification into the mBlock learning environment markedly enhanced students' motivation across all assessed dimensions: Attention, Relevance, Confidence, and Satisfaction.

The improved scores in Attention and Confidence suggested that the gamified elements made the programming concepts more appealing and accessible, helping to alleviate the initial apprehension or disinterest students might have felt toward programming. Similarly, the increased Relevance and Satisfaction scores indicated that students could see the practical utility and enjoyment in learning programming, potentially translating these skills into meaningful real-world applications. The use of gamification, therefore, not only mitigated students' negative attitudes toward programming but fostered a more profound recognition of programming's value, making the learning experience more engaging and rewarding.

5.3. RQ3) What Are the Outcomes in Terms of Student Achievement After Engaging with mBlock Through Gamification?

The overall correctness rate of 80.3% from the PAT underscored the students' diligent efforts to acquire computational thinking skills and to apply these principles in their programming tasks over the mBlock and gamification curriculum.

The completion rate for questions related to Sequence, Events, Data, and Operators exceeded 80%, while the completion rate for Parallelism, Conditions, and Loops was around 73%, indicating the need for further practice and instruction to facilitate a deeper understanding of these concepts.

5.4. RQ4) What Is the Relationship Between Computational Thinking, Learning Motivation, and Programming Achievement when Students Engage with mBlock Through Gamification?

The Pearson correlation test offered insights into the interdependencies among the constructs of computational thinking, learning motivation, and programming achievement.

The significant correlations within the PCTS dimensions demonstrated that the dimensions of computational thinking are closely interrelated, suggesting that efforts to improve one dimension might be associated with positive changes in the others.

Similarly, within the IMMS dimensions, the strong correlations suggested that the various aspects of motivation in learning are interconnected. This indicates that interventions that affect one motivational aspect might also influence others, impacting students' motivation toward learning programming.

 doi:10.20944/preprints202508.1903.v1

The significant correlations between the dimensions of PCTS and IMMS highlighted the relationship between computational thinking skills and motivational factors in programming education. This finding emphasized the association between having advanced computational thinking skills and experiencing positive motivational outcomes.

Finally, the correlation between PCTS, IMMS, and PAT emphasized the overarching relationship between computational thinking, motivation, and programming performance. Notably, this relationship underscored the association between students' confidence in their actual performance in programming achievements. While this finding suggested a strong association, it was necessary to note that other motivational and computational thinking dimensions showed correlations with PAT to varying extents, though not all reached statistical significance.

6. Conclusions

This study indicated notable enhancements in students' understanding of Computational Concepts and Computational Practices after the gamified experiment. However, the improvement in Computational Perspectives could have been more marked, suggesting that further instructional innovations may be needed. Additionally, the results indicated a substantial uplift in students' learning motivation across dimensions such as Attention, Relevance, Confidence, and Satisfaction post-intervention, highlighting the effectiveness of gamification in creating an engaging and stimulating learning environment.

Moreover, a notable correlation was identified between computational thinking and learning motivation, underscoring the correlation between advancements in computational thinking and students' motivational experiences. Furthermore, a significant correlation was found between students' confidence and programming performance, highlighting the comprehensive connection between computational thinking, motivation, and programming achievement.

Our study confirmed the potential of gamification as a powerful tool in the educational domain, particularly in primary school programming education for young learners. By making learning engaging, gamification improved students' understanding of programming concepts and their willingness and enthusiasm to learn.

Future work could include qualitative interviews with students to understand the educational impact of gamification in block-based programming more comprehensively. This would allow researchers to delve deeper into the subjective experiences, challenges, and motivational factors as perceived by the learners, providing a more nuanced and holistic view of the effectiveness and potential areas for refinement in the gamified learning environment.

References

1. A. Robins, J. Rountree, and N. Rountree, 'Learning and teaching programming: A review and discussion', *Computer science education*, vol. 13, no. 2, pp. 137–172, 2003.
2. Z. Zhan, L. He, Y. Tong, X. Liang, S. Guo, and X. Lan, 'The effectiveness of gamification in programming education: Evidence from a meta-analysis', *Computers and Education: Artificial Intelligence*, vol. 3, p. 100096, 2022, doi: 10.1016/j.caeai.2022.100096.
3. D. Weintrop, 'Block-based programming in computer science education', *Commun. ACM*, vol. 62, no. 8, pp. 22–25, July 2019, doi: 10.1145/3341221.
4. K. Brennan and M. Resnick, 'New frameworks for studying and assessing the development of computational thinking', in *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, 2012, p. 25.
5. T. Jenkins, 'On the difficulty of learning to program', in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Citeseer, 2002, pp. 53–58.
6. J. Du, H. Wimmer, and R. Rada, '" Hour of Code": Can It Change Students' Attitudes Toward Programming?', *Journal of Information Technology Education: Innovations in Practice*, vol. 15, p. 53, 2016.

7. B. Du Boulay, 'Some difficulties of learning to program', *Journal of Educational Computing Research*, vol. 2, no. 1, pp. 57–73, 1986.
8. J. Boustedt *et al.*, 'Threshold concepts in computer science: do they exist and are they useful?', *ACM Sigcse Bulletin*, vol. 39, no. 1, pp. 504–508, 2007.
9. J. L. Agapito, M. Rodrigo, and T. Mercedes, 'Designing an intervention for novice programmers based on meaningful gamification: An expert evaluation', 2017.
10. N. Zeybek and E. Saygı, 'Gamification in Education: Why, Where, When, and How?—A Systematic Review', *Games and Culture*, p. 155541202311586, Mar. 2023, doi: 10.1177/15554120231158625.
11. D. Maryono, S. Budiyono, and M. Akhyar, 'Implementation of Gamification in Programming Learning: Literature Review', *Int. J. Inf. Educ. Technol*, 2022.
12. W. C. Choi and I. C. Choi, 'Investigating the Effect of the Serious Game CodeCombat on Cognitive Load in Python Programming Education', in *2024 IEEE World Engineering Education Conference (EDUNINE)*, IEEE, 2024, pp. 1–6. doi: 10.1109/EDUNINE60625.2024.10500551.
13. D. Pérez-Jorge and M. C. Martínez-Murciano, 'Gamification with Scratch or App Inventor in Higher Education: A Systematic Review', *Future Internet*, vol. 14, no. 12, p. 374, 2022.
14. F. J. Agbo, S. S. Oyelere, J. Suhonen, and T. H. Laine, 'Co-design of mini games for learning computational thinking in an online environment', *Education and Information Technologies*, vol. 26, no. 5, pp. 5815–5849, 2021.
15. A. K. Yadav and S. S. Oyelere, 'Contextualized mobile game-based learning application for computing education', *Education and Information Technologies*, vol. 26, pp. 2539–2562, 2021.
16. A. Baytak and S. M. Land, 'An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom', *Educational Technology Research and Development*, vol. 59, pp. 765–782, 2011.
17. K. Gross and S. Gross, 'Transformation: Constructivism, design thinking, and elementary STEAM', *Art Education*, vol. 69, no. 6, pp. 36–43, 2016.
18. Y. B. Kafai and Q. Burke, 'Constructionist gaming: Understanding the benefits of making games for learning', *Educational psychologist*, vol. 50, no. 4, pp. 313–334, 2015.
19. J. J. Bullon, A. H. Encinas, M. J. S. Sanchez, and V. G. Martinez, 'Analysis of student feedback when using gamification tools in math subjects', in *2018 IEEE Global Engineering Education Conference (EDUCON)*, Tenerife: IEEE, Apr. 2018, pp. 1818–1823. doi: 10.1109/EDUCON.2018.8363455.
20. V. Panthalookaran, 'Gamification of physics themes to nurture engineering professional and life skills', in *2018 IEEE Global Engineering Education Conference (EDUCON)*, Tenerife: IEEE, Apr. 2018, pp. 931–939. doi: 10.1109/EDUCON.2018.8363330.
21. A. H. S. Metwally and W. Yining, 'Gamification in massive open online courses (MOOCs) to support chinese language learning', in *2017 International Conference of Educational Innovation through Technology (EITT)*, IEEE, 2017, pp. 293–298.
22. D. Dicheva, C. Dichev, G. Agre, and G. Angelova, 'Gamification in education: A systematic mapping study', *Journal of educational technology & society*, vol. 18, no. 3, pp. 75–88, 2015.
23. S. de Sousa Borges, V. H. Durelli, H. M. Reis, and S. Isotani, 'A systematic mapping on gamification applied to education', in *Proceedings of the 29th annual ACM symposium on applied computing*, 2014, pp. 216–222.
24. I. Caponetto, J. Earp, and M. Ott, 'Gamification and education: A literature review', in *European Conference on Games Based Learning*, Academic Conferences International Limited, 2014, p. 50.
25. M. F. Areed, M. A. Amasha, R. A. Abougalala, S. Alkhalaf, and D. Khairy, 'Developing gamification e-quizzes based on an android app: the impact of asynchronous form', *Education and Information Technologies*, vol. 26, pp. 4857–4878, 2021.
26. M. L. Vaquero, 'Gamificación y juegos serios en Educación Superior.', in *IV Congreso Virtual Internacional sobre Innovación Pedagógica y Praxis Educativa INNOVAGOGÍA 2018: libro de actas. 20, 21 y 22 de marzo 2018*, AFOE. Asociación para la Formación, el Ocio y el Empleo, 2018, p. 100.
27. J. M. Wing, 'Computational thinking', *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006, doi: 10.1145/1118178.1118215.

28. D. Hemmendinger, 'A plea for modesty', *ACM Inroads*, vol. 1, no. 2, pp. 4–7, June 2010, doi: 10.1145/1805724.1805725.
29. W. C. Choi, I. C. Choi, and C. I. Chang, 'Improving computational thinking through flipped classroom: A case study on K-12 programming course in Macao.', *International Research Journal of Science, Technology, Education, & Management (IRJSTEM)*, vol. 4, no. 3, 2024.
30. W. C. Choi and C. I. Chang, 'The Students' Perspective on Computational Thinking through Flipped Classroom in K-12 Programming Course', in *Proceedings of the 2024 8th International Conference on Education and Multimedia Technology*, 2024, pp. 106–113.
31. W. C. Choi, 'The Influence of Code.org on Computational Thinking and Learning Attitude in Block-Based Programming Education', in *Proceedings of the 2022 6th International Conference on Education and E-Learning*, Yamanashi Japan: ACM, Nov. 2022, pp. 235–241.
32. W. C. Choi and I. C. Choi, 'The Influence and Relationship between Computational Thinking, Learning Motivation, Attitude, and Achievement of Code.org in K-12 Programming Education', *arXiv preprint arXiv:2412.14180*, 2024.
33. C. Aldrich, *Learning online with games, simulations, and virtual worlds: Strategies for online instruction*. John Wiley & Sons, 2009.
34. J. D. Bayliss and S. Strout, 'Games as a" flavor" of CS1', in *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, 2006, pp. 500–504.
35. W. C. Choi and I. C. Choi, 'Discovering Programming Talent and Improving Learning Motivation with CodeCombat in K-12 Education', in *2024 IEEE World Engineering Education Conference (EDUNINE)*, IEEE, 2024, pp. 1–6. doi: 10.1109/EDUNINE60625.2024.10500458.
36. B. Huang and K. F. T. Hew, 'Measuring learners' motivation level in massive open online courses', *International Journal of Information and Education Technology*, 2016.
37. K. M. Law, S. Geng, and T. Li, 'Student enrollment, motivation and learning performance in a blended learning environment: The mediating effects of social, teaching, and cognitive presence', *Computers & Education*, vol. 136, pp. 1–12, 2019.
38. C. Kazimoglu, 'Enhancing confidence in using computational thinking skills via playing a serious game: A case study to increase motivation in learning computer programming', *IEEE Access*, vol. 8, pp. 221831–221851, 2020.
39. S. Leutenegger and J. Edgington, 'A games first approach to teaching introductory programming', in *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, 2007, pp. 115–118.
40. T. Barnes, H. Richter, E. Powell, A. Chaffin, and A. Godwin, 'Game2Learn: building CS1 learning games for retention', in *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, 2007, pp. 121–125.
41. M. Muratet, P. Torguet, F. Viallet, and J.-P. Jessel, 'Experimental feedback on Prog&Play: a serious game for programming practice', in *Computer graphics forum*, Wiley Online Library, 2011, pp. 61–73.
42. J. M. Keller, 'Motivation and instructional design: A theoretical perspective', *Journal of instructional development*, pp. 26–34, 1979.
43. Ş. Büyüköztürk, E. Kılıç Çakmak, Ö. E. Akgün, Ş. Karadeniz, and F. Demirel, *Scientific research methods*, 3rd edn. Ankara: PegemA, 2008.
44. Classdojo, 'What's ClassDojo?' [Online]. Available: classdojo.com/about/
45. P. Z. Wu, 'Compilation of Computational Thinking Self-efficacy Scale and Verification of Reliability and Validity', Taichung University of Education, Taichung, Taiwan, 2021.
46. B. Zhong, Q. Wang, J. Chen, and Y. Li, 'An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking', *Journal of Educational Computing Research*, vol. 53, no. 4, pp. 562–590, Jan. 2016, doi: 10.1177/0735633115608444.
47. J. M. Keller, 'The Arcs model of motivational design', *Motivational design for learning and performance: The ARCS model approach*, pp. 43–74, 2010.

doi:10.20944/preprints202508.1903.v1

disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.