Game-Based Learning in Programming Courses: A Platform Development for Enhanced Learning Engagement

Noven Indra Prasetya* and Nia Saurina

Department of Informatics, Faculty of Engineering, Universitas Wijaya Kusuma Surabaya, Surabaya, Indonesia
Email: noven@uwks.ac.id (N.I.P.); niasaurina@uwks.ac.id (N.S.)
*Corresponding author

Manuscript received February 24, 2025; revised May 13, 2025; accepted June 20, 2025; published December 9, 2025

***Abstract*—This study aims to develop a game-based learning platform specifically designed to enhance student engagement and learning outcomes in programming courses. Utilizing a Design-Based Research approach and the Analysis, Design, Development, Implementation, Evaluation (ADDIE) model, the platform integrates game elements such as live coding, progressive challenges, experience points, leaderboards, and badges to create an interactive and motivating learning environment. The study involved 276 students enrolled in a structured programming course, with data collected through pre-tests, post-tests, and engagement questionnaires. The analysis revealed that the platform significantly improved student engagement, with an average increase of 208.62 points, and learning outcomes, with an average improvement of 19.94 points. A paired sample t-test confirmed a highly significant difference between pre-test and post-test scores, supporting the platform's effectiveness in enhancing students' programming skills. This game-based platform not only successfully boosted engagement and learning outcomes but also contributed significantly to innovation in programming learning. The study recommends further development to support other programming languages and broader user populations, as well as exploration of artificial intelligence integration to enrich the learning experience.**

***Keywords*—game-based learning, learning platform, programming learning, student engagement, learning outcomes**

I. Introduction

The rapid advancement of information technology has significantly influenced educational methods, especially in technical fields such as programming. Students often struggle with abstract and complex programming concepts, which leads to low engagement and achievement [1]. In response, Game-Based Learning (GBL) has emerged as a promising pedagogical approach, offering interactive and motivating environments that can potentially transform traditional programming instruction.

Despite its potential, the implementation of GBL in programming education still faces notable challenges. Most platforms do not adequately support essential programming competencies such as algorithmic thinking and debugging. Furthermore, limited integration of real-time coding features and motivational game elements, like leaderboards and achievement systems, reduces the capacity of such platforms to sustain long-term engagement [2].

Although previous studies have explored various applications of Game-Based Learning (GBL) in computer science education, most have focused on surface-level gamification (e.g., quizzes, badges, or point systems) without integrating core programming practices such as real-time coding, iterative problem-solving, or adaptive challenge progression. Moreover, few platforms have been designed and evaluated using a rigorous instructional design framework such as ADDIE within a Design-Based Research (DBR) approach.

This study addresses that gap by developing and validating a GBL platform that embeds pedagogically aligned game mechanics into the structured learning of C++ programming, aiming to enhance both student engagement and learning performance through theoretically grounded, evidence-based design. The novelty of this research lies in the platform's design, which is not only tailored to the needs of programming learning but also intended to enhance algorithmic logic skills and the ability to solve complex problems. The scope of the study includes the development, testing, and evaluation of the platform to assess its impact on student engagement and learning outcomes in programming courses. The study is grounded in the Design-Based Research methodology and employs the ADDIE model to guide development. Thus, this research is expected to make a significant contribution to the GBL literature, particularly in the context of programming learning.

II. Literature Review

A. Game-Based Learning in Programming Learning

Game-Based Learning (GBL) has evolved into an increasingly adopted approach in education, particularly in the field of programming. GBL refers to the integration of game elements into the learning process to enhance student engagement and motivation [3]. Previous studies have demonstrated that game-based approaches can improve students' understanding of programming concepts and problem-solving skills [4, 5]. In the context of Science, Technology, Engineering, and Mathematics (STEM) education, including programming, GBL has been proven to assist students in overcoming cognitive challenges by providing a more interactive and engaging learning environment [6].

One of the key factors in the success of GBL in programming learning is gamification, which involves the application of game mechanics such as points, badges, and leaderboards in the learning process [7, 8]. Studies by Lee *et al.* and Zaki *et al.* [9, 10] have shown that incorporating gamification in programming courses can enhance learning outcomes through instant feedback and rewards for student achievements. Additionally, gamification elements can foster collaboration and healthy competition among students, creating a more dynamic and productive learning community [11, 12].

doi: 10.18178/ijiet.2025.15.12.2455

B. Trends and Challenges in Implementing GBL for Programming

Although the potential of GBL in programming learning is highly promising, its implementation still faces several challenges. One of the primary obstacles is the lack of GBL platforms specifically designed to support students' algorithmic logic and debugging skills [13]. Most existing GBL platforms have yet to effectively integrate live coding features and real-time student engagement monitoring mechanisms, which may reduce their effectiveness in enhancing learning motivation [14]. Additionally, limitations in game design can negatively impact the effectiveness of GBL as a learning tool [10, 15].

Several studies emphasize that an effective GBL platform should integrate elements of competition, collaboration, and narrative that are relevant to the context of programming learning [16]. The integration of live coding has been proven to offer significant benefits by allowing students to observe the immediate impact of the code they write, enhancing their debugging skills, and deepening their understanding of algorithmic concepts [5, 15]. However, most existing research remains limited to the implementation of simple games without accommodating the complexities of programming, such as code completion through live coding, debugging features, and leaderboards that can drive higher levels of engagement [10].

III. Materials and Methods

A. Research Objectives, Questions, and Hypotheses

The objective of this study is to design, implement, and evaluate a Game-Based Learning (GBL) platform aimed at enhancing student engagement and improving learning outcomes in programming learning.

This study is guided by the following research questions:

1) Does the use of the GBL platform significantly improve student engagement in programming courses?
2) Does the GBL platform significantly enhance student learning outcomes in terms of programming skills?

Based on these questions, the study proposes the following research hypotheses:

1) H_1: There is a statistically significant increase in student engagement after the use of the game-based learning platform.
2) H_2: There is a statistically significant improvement in students' programming learning outcomes after using the platform.

B. Research Design and Approach

This study adopts a **Design-Based Research (DBR)** methodology as defined by Ulfa *et al.* [17], which emphasizes iterative, collaborative, and context-sensitive interventions in real educational environments. To structure the design and development of the learning solution, the study employed the **ADDIE model** (Analysis, Design, Development, Implementation, and Evaluation) [18], embedded within the DBR's iterative cycles. The integration of DBR and ADDIE ensured that the platform was designed systematically while remaining responsive to continuous evaluation and user feedback.

The following describes how the ADDIE phases were aligned with the DBR framework.

1) Analysis Phase (DBR: Preparation and Planning): the research began with a needs analysis involving curriculum review, identification of student challenges in learning programming, and analysis of engagement levels using pre-surveys. The analysis also examined technological opportunities for integrating game elements with educational goals.
2) Design Phase (DBR: Designing a Solution): in this phase, the platform's structure and learning mechanics were conceptualized. Game elements—such as live coding, progressive levels, badges, and leaderboards—were mapped to learning objectives in the C++ programming curriculum. A wireframe and instructional flow were designed to align gamified features with cognitive, behavioral, and emotional engagement principles.
3) Development Phase (DBR: Iteration Cycle 1): the initial prototype was developed based on the design specifications. Using black-box testing, system functionality was validated. A small group of students tested the prototype, and their feedback was collected to identify usability issues, instructional misalignments, and engagement obstacles. This phase concluded with revisions to the platform structure.
4) Implementation Phase (DBR: Iteration Cycle 2): the improved version of the platform was deployed in a broader classroom context over five weeks. Students engaged in learning activities using the platform, and data were collected via post-tests, engagement questionnaires, and user satisfaction surveys. Real-time feedback and leaderboard features were monitored to analyze learner behaviors.
5) Evaluation Phase (DBR: Analysis of Results): quantitative data were analyzed using descriptive and inferential statistics (paired sample t-tests) to evaluate the effectiveness of the intervention. This phase confirmed improvements in both student engagement and learning outcomes, and informed recommendations for future platform iterations.

This structured yet iterative process ensured that the developed solution evolved through practical trials while remaining aligned with pedagogical objectives. The detailed results of each ADDIE phase within the DBR cycles are presented in result and discussion section.

The ADDIE model was selected over other instructional design frameworks because of its structured, iterative nature and its proven adaptability in technology-enhanced learning environments, including serious games and game-based learning platforms. Unlike models such as Kemp or Dick & Carey, which emphasize linear or content-centric approaches, ADDIE provides a clear, cyclical process that aligns well with the needs of iterative platform development in a DBR context. Its phases—Analysis, Design, Development, Implementation, and Evaluation—mirror the design-thinking process often used in educational game development, allowing for continuous feedback integration, user testing, and pedagogical refinement.

C. Participants

The participants in this study consisted of 276 undergraduate students enrolled in a Structured Programming course at Universitas Wijaya Kusuma Surabaya. All students were selected using purposive sampling, as outlined by Campbell *et al.* [19], focusing on individuals who were actively engaged in programming coursework and had similar baseline exposure to C++ programming concepts. The participant cohort represented a diverse set of students in terms of initial programming proficiency and learning motivation, making it suitable for evaluating the effectiveness of the GBL platform in varied learner contexts.

D. Data Collection and Analysis

This study employed three primary data collection instruments: (1) pre- and post-tests of programming knowledge, (2) a student engagement questionnaire, and (3) a user satisfaction questionnaire. All instruments were developed or adapted based on established frameworks and were evaluated for content validity and reliability prior to use.

1) Knowledge Tests (Pre-Test and Post-Test): the knowledge tests consisted of 20 multiple-choice and short-answer items that covered core programming concepts in C++, including control structures, loops, functions, and arrays. These items were constructed based on course learning outcomes and reviewed by two programming instructors for content validity. The reliability of the test was assessed using **Cronbach's alpha**, which yielded a value of **0.82**, indicating good internal consistency.
2) Student Engagement Questionnaire: this instrument was adapted from a widely used student engagement scale in digital learning environments, comprising 15 items spanning three dimensions: cognitive engagement, emotional engagement, and behavioral engagement. Items were rated on a 5-point Likert scale. The instrument's content validity was ensured through expert judgment, and a pilot test involving 40 students produced a Cronbach's alpha of 0.88, confirming high reliability.
3) User Satisfaction Questionnaire: the user satisfaction questionnaire was based on the End-User Computing Satisfaction (EUCS) framework [20], which includes five dimensions: content, format, accuracy, timeliness, and ease of use. Each dimension was measured using 3 items, rated on a 5-point Likert scale. The instrument demonstrated strong internal consistency (Cronbach's alpha = 0.86) and was validated through a panel of instructional technology experts for relevance and clarity.

These psychometric evaluations confirm that the instruments used in this study are both valid and reliable, supporting the robustness of the findings related to student engagement, satisfaction, and learning outcomes.

The data were analyzed using quantitative statistical methods:

1) Descriptive statistics (mean, standard deviation, minimum, maximum, and quartiles) were used to summarize engagement scores and learning outcomes.
2) A paired sample t-test was applied to compare pre-test and post-test scores for both engagement and learning outcomes. This test evaluated whether observed changes were statistically significant, thus testing the proposed hypotheses (H_1 and H_2).
3) Additionally, visualizations such as box plots and score distribution charts were used to further illustrate shifts in engagement and performance metrics.

This approach ensured a robust, evidence-based assessment of the platform's pedagogical impact.

IV. RESULT AND DISCUSSION

A. Analysis

The game-based programming learning platform developed in this study is a web-based system specifically designed to support the learning process of the C++ programming language through an interactive, engaging, and immersive approach. This platform adopts a game-based concept with the primary objective of enhancing student engagement in understanding and applying structured programming concepts. A unique feature of the platform is the ability for students to select a character as their avatar, representing their identity while completing various missions within the game. Each mission or challenge is designed to be solved through live coding, a process of writing code directly, with instructions specifically tailored to develop programming logic and skills. The game elements incorporated into this platform are grounded in three key principles of GBL: fostering student engagement through an interactive learning environment, providing autonomy within a safe exploratory setting, and offering clear mechanisms for monitoring and reflecting on learning progress. These principles are implemented to create a more meaningful and effective learning experience.

Table 1. Game fundamental principle

Fundamental Principle	Contextual Principle	System Implementation
Engagement	Challenge	Students, acting as players within this platform, will encounter various problems or cases that require solutions through the application of their programming skills.
	Space Retrieval	The practice feature is a crucial element in supporting players to apply the concepts they have learned through the process of coding. Through the provided guidance, players can receive immediate feedback to refine or reinforce their understanding. Additionally, supplementary hints are available to assist players who encounter difficulties in completing tasks, ensuring a more supportive and directed learning experience.
	Character	Each player is provided with an avatar that they can control interactively to explore various aspects of the game, enabling a more personalized and immersive learning experience.
	Retrieval Practice	Players have access to comprehensively designed learning content, allowing them to revisit previously learned material and apply their understanding through interactive modules that support active and practical learning.
Autonomy & Safe Environment	Minimal Control	Players are granted autonomy and freedom to manage their learning process according to their individual preferences. However, given that programming is a hierarchical subject heavily reliant on the mastery of prior concepts, players are required to complete the game content sequentially to ensure a deep and continuous understanding.

Fundamental Principle	Contextual Principle	System Implementation
Progression	Earning Point	Players earn points as rewards each time they successfully complete the material at each level, serving as additional motivation to enhance engagement and active participation in the learning process.
	Earning Achievement	The platform includes a badge system that provides information on levels, the steps required to earn each badge, and the number of users who have achieved them, offering additional motivation for players to reach their learning goals.

Table 1 outlines the core game elements integrated into the development of the GBL platform. These design choices are informed by established theories in both educational psychology and game-based learning. For example, the inclusion of challenge, feedback, and progression mechanisms aligns with constructivist learning theory [21], which emphasizes learning through active problem-solving and iterative feedback. The use of points, badges, and leaderboards supports self-determination theory [22], particularly in fostering intrinsic motivation through the fulfillment of competence, autonomy, and relatedness.

Furthermore, the incorporation of immersive narrative and incremental task complexity reflects principles from flow theory [23], aiming to maintain an optimal balance between skill and challenge to sustain learner engagement. These theoretical connections provide a robust foundation for the platform's design, ensuring that game mechanics are not arbitrarily implemented but pedagogically aligned with student learning processes.

B. Design

To reflect the iterative nature of the DBR process, this section describes the two core design cycles implemented during the development of the platform, as shown in Fig. 1. Each cycle included user testing, system validation, and refinement of both pedagogical features and technical functionality.

Fig. 1. Game-based platform interface.

Table 2. System testing

Module	Expectations	Result
Material	This platform is designed to comprehensively present C++ programming learning materials, enabling players to easily access and understand key concepts.	Valid
Experience Points	The platform is equipped with the ability to calculate experience points based on the difficulty levels successfully completed by players, providing proportional rewards for their achievements in the learning process.	Valid
Progression	The system is designed to calculate player progress as a percentage, allowing players to track how much of the learning material they have completed overall.	Valid

In the first cycle, a basic prototype of the platform was deployed in a limited classroom setting. This version focused on integrating key gamification elements, such as live coding, experience points, and progressive challenges. Black-box testing was conducted to ensure the core features met the defined functional specifications (see Table 2). Additionally, 10–15 students participated in exploratory sessions, providing feedback on navigation, content delivery, and engagement levels. Key feedback from this phase led to improvements in the interface layout and the feedback system for coding tasks.

The second cycle involved deploying the refined version of the platform in a broader context, incorporating updates based on the initial feedback. A more detailed user satisfaction survey was administered, drawing on the EUCS framework, and results showed high levels of student satisfaction across five evaluated aspects (Table 3). Additional improvements were made to the adaptive feedback mechanism, avatar personalization, and sequencing of coding challenges.

Table 3. User testing

Aspect	Number of Respondents	Min	Max	Mean	Standard Deviation	Level	Result
Accuracy	15	2	5	4.17	0.62	5	Very Strong
Content	15	1	5	4.22	0.64	5	Very Strong
Ease of Use	15	2	5	4.48	0.64	5	Very Strong
Format	15	1	5	4.32	0.50	5	Very Strong
Timeliness	15	2	5	4.31	0.50	5	Very Strong

These iterative cycles were essential in aligning the platform's design with user needs and pedagogical goals, demonstrating the value of the DBR methodology in producing a practical, evidence-based solution for programming learning.

C. Development

The platform was built using a modular client-server architecture. The frontend was developed using HyperText Markup Language 5 (HTML5), Cascading Style Sheets (CSS), and JavaScript with Vue.js for interactive components. The backend was implemented using Unity to handle Application Programming Interface (API) requests, user authentication, and game logic. A My Structured Query Language (MySQL) database was used to store user progress, code submissions, feedback records, and leaderboard data. The code execution engine was containerized using Docker to securely run user-submitted C++ code in an isolated environment. The platform also included an admin panel for instructors to manage challenges, track student progress, and export analytics. This architecture was chosen for scalability, cross-platform accessibility, and real-time responsiveness to user input.

During the development phase, the platform underwent limited testing to ensure its quality and functionality prior to broader-scale trials. Testing was conducted through two main approaches: system testing and user satisfaction evaluation. System testing employed the black box testing method, which focuses on verifying the fulfillment of all functional system requirements without examining the internal implementation details. This method ensures that each component of the platform operates in accordance with the specified design and requirements. Meanwhile, user satisfaction evaluation involved a group of students assessing key aspects of the platform, such as content quality, layout, navigation ease, timeliness, and the accuracy of material presentation. The evaluation utilized the End-User Computing Satisfaction (EUCS) instrument, which has been empirically validated as developed by Purwanto and Hedin [20]. This instrument was designed to provide valid and reliable data on users' satisfaction and experience with the platform. The results serve as a foundation for further improvements and development to ensure the platform remains responsive to user needs.

The system testing results, presented in Table 2, demonstrate comprehensive validation of the fulfillment of all functional requirements using the black box testing method. This testing focused on critical aspects of the system, such as the accuracy of learning material presentation, consistency in experience point calculations, and the recording and management of player progress. The method ensured that each platform function operates as designed, without technical errors or inconsistencies that could affect user experience or reduce system reliability. Consequently, the testing outcomes not only verified the platform's technical performance but also confirmed that the system is capable of optimally supporting the learning process and meeting user needs. This validation serves as a crucial step in ensuring system integrity before proceeding to broader-scale trials.

User testing involved 10–15 students participating in a five-week exploratory class, during which they studied programming materials and completed challenges through the platform. User satisfaction data, presented in Table 3, were collected using a questionnaire-based instrument with structured variable questions. Descriptive analysis of the data indicated an average user satisfaction level above 4.00 on a Likert scale, reflecting highly positive responses to the platform. These findings align with previous research by Pradana *et al.* [24], which reported high satisfaction levels with similar platforms utilizing the GBL approach. The high satisfaction rates suggest that the platform is not only well-received by users but also holds significant potential for broader integration into learning processes, as it is perceived to effectively support interactive and meaningful learning experiences.

D. Implementation

The implementation phase is a critical step in integrating the platform into the programming learning process in the classroom, with the primary goal of enhancing student engagement while optimizing learning outcomes through an interactive and innovative approach. During the five-week implementation period, students directly interacted with various game features on the platform, such as live coding challenges, leaderboards, and a level system that must be completed progressively. These features were designed not only to create an enjoyable learning environment but also to motivate students to actively engage in the learning process. Direct interaction with these elements provided a learning experience that was not only engaging but also challenging, allowing students to deepen their understanding of programming concepts effectively. This approach is expected to foster deeper engagement and improve learning success through a method that integrates cognitive, emotional, and behavioral aspects.

In both the initial and final surveys, student engagement was measured using a 15-item questionnaire, with each item rated on a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). The total engagement score for each student was calculated as the sum of all item responses, resulting in a maximum possible score of 75 points. To classify students into "high" and "low" engagement categories, a threshold score of ≥60 (i.e., an average item score of 4 or higher) was used to indicate high engagement. Students scoring below 60 were categorized as having low engagement. This operational definition aligns with established engagement scales and reflects consistently positive responses across emotional, behavioral, and cognitive dimensions of engagement. This binary classification enabled straightforward comparison of engagement levels before and after the intervention.

This survey involved 276 students, with the results summarized in Table 4, providing a baseline overview of student engagement patterns in programming learning. The data encompass aspects such as active participation, interest in the material, and interaction with learning resources. These initial survey results serve as a foundation for evaluating the effectiveness of the GBL platform by comparing changes in student engagement levels after they began interacting with the platform during the learning process. This comparative analysis is expected to provide empirical evidence of the platform's positive impact on enhancing overall student engagement.

Table 4. Initial student engagement data summary

Engagement Level	Number of Students
High	124
Low	152

Table 4 presents the distribution of student engagement levels based on the initial survey results, categorized into high and low engagement levels as determined by survey responses. A total of 124 students were classified as having high engagement, with all their responses scoring 4 or 5. This indicates consistent levels of interest, participation, and activity in learning activities, as reflected in the survey indicators covering cognitive, emotional, and behavioral aspects. Conversely, 152 students fell into the low engagement category, as one or more of their responses scored below 4. This suggests that students in this group exhibited suboptimal engagement, potentially due to a lack of motivation, less engaging teaching methods, or other obstacles in the learning process. These findings highlight the need for adopting more interactive and innovative learning approaches, such as the use of GBL platforms, to enhance engagement and facilitate more enjoyable and meaningful learning experiences.

Out of the 276 students who participated in the survey, approximately 45% demonstrated high engagement levels, while the remaining 55% were categorized as having low engagement. This proportion indicates that the majority of students have yet to reach the expected level of engagement in the learning process. These findings are significant, as low engagement can negatively impact the effectiveness of learning and the achievement of learning outcomes. The data provide a foundation for evaluating the current teaching methods and highlight the need for strategic interventions to enhance student engagement. Interventions such as integrating interactive and innovative GBL platforms are expected to motivate students to participate more actively in the learning process. Consequently, this approach aims not only to improve engagement levels but also to foster more optimal and meaningful learning outcomes.

The final survey was designed to evaluate student engagement levels after the implementation of the GBL platform as an intervention. The data summarized in Table 5 show significant changes in student engagement compared to the initial survey. This survey assessed cognitive, emotional, and behavioral aspects, reflecting students' active participation and motivation in the learning process. The analysis of the survey results provides empirical insights into the platform's effectiveness in enhancing engagement, focusing on increased participation in learning challenges, motivation to reach higher levels, and more meaningful interactions with programming materials. These findings reinforce that the use of game-based platforms can serve as a strategic solution to address low engagement and create more interactive and engaging learning experiences. Consequently, these results provide a foundation for further development in integrating game-based technologies into programming learning methods.

Table 5. Final student engagement data summary

Engagement Level	Number of Students
High	248
Low	28

The results of the final survey reveal that 248 out of 276 students (approximately 89%) demonstrated high engagement levels, scoring 4 or 5 on every statement in the questionnaire. The post-intervention data show that 89% of students reported high engagement, confirming the platform's effectiveness. The relatively small proportion of low-engagement students compared to those with high engagement highlights the success of the intervention in creating a more engaging and meaningful learning experience. These findings suggest that the GBL platform significantly enhances overall student engagement in terms of participation, motivation, and interaction with learning materials. This success underscores the potential of the platform as an innovative method for programming learning, fostering more optimal learning outcomes.

In addition to the quantitative results, qualitative responses from students provided deeper insight into their experiences with the platform. Many students described the platform as "motivating," "fun," and "different from traditional learning." Several highlighted that the immediate feedback during coding tasks helped them better understand their mistakes, while others appreciated the progress bar and avatar system as "visual indicators of growth." A recurring theme was the sense of autonomy and control over their learning pace, with students noting that they felt "less pressure" and "more freedom to experiment." These qualitative insights align with the platform's intended design and reinforce its impact on both engagement and motivation.

The high number of students with elevated engagement levels indicates the success of the GBL platform in creating an interactive and relevant learning environment. Survey responses indicate that students perceived challenges (+1.4), competitive features (+1.2), and reward mechanisms as key motivators. These elements appear to have directly contributed to the observed increase in engagement.

Additionally, the relevance of the material to students' learning needs, coupled with the engaging activity format, plays a crucial role in fostering deep engagement. This approach not only enhances students' interest in the topics being taught but also encourages their active participation in the learning process, creating a more enjoyable learning experience while effectively supporting the achievement of learning objectives.

While the increase in high engagement levels—from 45% to 89%—demonstrates the potential effectiveness of the GBL platform, it is important to consider possible confounding variables that may have influenced these outcomes. For example, improvements could partially result from novelty effects associated with the introduction of new technology, or the increased attention given to students during the intervention period (i.e., the Hawthorne effect). Additionally, external factors such as students' prior familiarity with gamified applications or parallel support from instructors may have contributed to the observed results. Although these factors were not directly measured in this study, their potential influence underscores the importance of interpreting results within the broader context of the classroom environment. Future research using control groups or mixed-methods approaches could help isolate the platform's effects more precisely and provide a deeper understanding of the mechanisms driving student engagement.

E. Evaluation

The evaluation of the platform's effectiveness was conducted by collecting quantitative data on student engagement levels and learning outcomes throughout the learning process using the game-based platform. The data were analyzed using descriptive statistical methods to provide an overview of the distribution and characteristics of student engagement and learning outcomes. Additionally, inferential analysis was conducted through a paired sample t-test to determine significant differences in programming skills or learning outcomes before and after using the platform. This analytical approach aims to provide empirical evidence of the platform's effectiveness in enhancing student engagement and academic achievement. Thus, the evaluation not only assesses the success of the platform's implementation but also offers critical insights into the contribution of the GBL approach to the quality of programming learning.

Table 6. Descriptive statistics of engagement scores

Aspect	Initial Score	Final Score
count	24	24
mean	1079.708	1288.333
std	11.351	6.742
min	1047	1275
25%	1073	1285.5
50%	1079.5	1289
75%	1087.25	1292
max	1097	1303

Table 6 presents engagement scores based on a validated engagement questionnaire. Each student responded to 15 Likert-scale items (1–5), covering emotional, cognitive, and behavioral engagement. The total score (max 1500) reflects the aggregate across these items.

The descriptive statistical analysis indicates a significant increase in student engagement following the use of the game-based platform. The average engagement score in the initial survey was 1079.71 with a standard deviation of 11.35, while the final survey showed an increased average score of 1288.33 with a standard deviation of 6.74. The mean increase of 208.62 reflects a substantial improvement in student engagement throughout the learning process. Furthermore, the tighter distribution of final scores, as evidenced by the reduced standard deviation, suggests that the increase in engagement was more consistent across the student group. These findings provide strong evidence that the game-based platform is not only effective in enhancing overall student engagement but also fosters a more equitable and inclusive learning experience among participants.

The analysis using a paired sample t-test revealed a t-statistic value of −69.67 with a p-value of 2.91×10^{-28}, which is significantly below the 0.05 significance threshold. These results indicate a statistically significant difference in student engagement scores before and after the use of the game-based platform. Consequently, the null hypothesis (H_0), which states that there is no mean difference, can be rejected. These findings provide empirical evidence that the game-based platform intervention has a significant positive impact on enhancing student engagement. The data support the claim that the GBL approach is effective in creating a more engaging and productive learning experience while simultaneously improving the overall quality of the learning process.

In addition to statistical significance, the practical significance of the findings was assessed using Cohen's d, a standardized measure of effect size. For engagement scores, the mean difference of 208.62 points yielded a Cohen's d = 2.63, indicating a very large effect. For learning outcomes, the improvement of 19.94 points resulted in a Cohen's d = 7.10, which also represents an extremely large effect size. According to commonly accepted benchmarks Cohen [25], a d-value above 0.8 is considered large; therefore, these values highlight the substantial practical impact of the GBL platform on both student engagement and programming achievement.

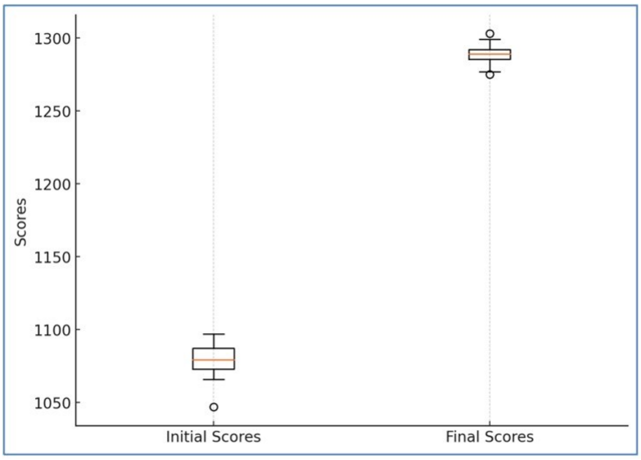

Fig. 2. Comparison of engagement scores.

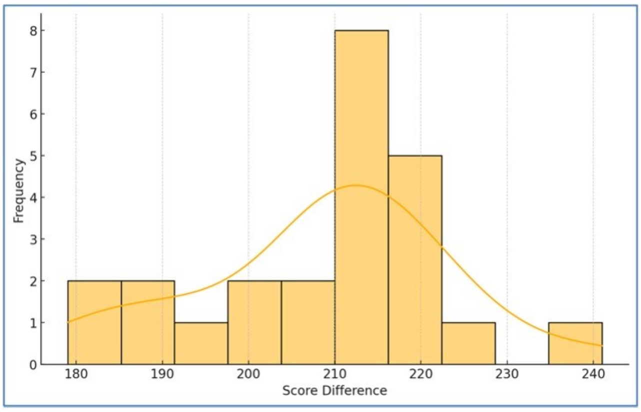

Fig. 3. Distribution of engagement scores.

Fig. 2 displays the distribution of student engagement scores from the pre-intervention (initial) and post-intervention (final) surveys. The x-axis represents individual students (anonymized), and the y-axis shows their total engagement score (ranging from 1000 to 1300 points, based on the composite of 15 Likert-scale items). The final distribution shows a clear upward shift, indicating increased engagement following the use of the game-based learning platform. Additionally, the score difference distribution diagram in Fig. 3 illustrates the difference in individual engagement scores before and after the platform intervention. The x-axis represents the range of score changes (e.g., 180–200, 200–220, etc.), while the y-axis shows the number of students in each category. The positive skew of the distribution indicates that most students experienced substantial engagement gains, with the majority falling in the 200–220 point increase range.

The significant increase in engagement scores reflects the success of the game-based platform intervention implemented in this study. Students exhibited higher emotional, cognitive, and behavioral engagement following the platform's implementation, as evidenced by an average score increase of 208.62 points. This improvement indicates that the platform effectively captured students' attention and enhanced their active participation in the learning process. These findings not only reaffirm the effectiveness of the GBL approach in fostering more interactive education but also provide strong empirical support for further development and integration of similar platforms into learning processes. With its potential to maximize student engagement, this platform can significantly contribute to improving the quality of learning and future academic outcomes.

Table 7. Descriptive statistics of learning outcomes

Aspect	Initial Score	Final Score
count	276	276
mean	69.728	89.667
std	2.648	2.833
min	62	73
25%	68	88
50%	70	90
75%	72	92
max	77	97

Table 7 presents learning outcomes based on pre- and post-test scores (max 100 points). These tests measured comprehension of core C++ programming concepts (e.g., control structures, arrays, functions), aligned with course learning outcomes.

The descriptive statistical analysis of student skills and learning outcomes, shows a significant improvement following the use of the game-based platform. The average initial student score was 69.73 with a standard deviation of 2.65, while the final average score increased to 89.67 with a standard deviation of 2.83, reflecting a mean improvement of 19.94 points. This increase demonstrates the platform's effectiveness in substantially enhancing students' programming skills. The narrower distribution of initial scores compared to the final scores indicates that, while most students experienced improvement, there was greater individual variation in the final outcomes. This variation suggests that the game-based platform's impact may differ among individuals, potentially influenced by factors such as initial engagement levels, prior experience, or learning style preferences. Nevertheless, these findings overall reinforce evidence that the game-based platform positively contributes to improving programming skills and serves as an innovative and effective approach to programming learning.

The paired sample t-test analysis of student learning outcomes revealed a t-statistic value of −83.36 with a p-value of 3.27×10^{-197}. This extremely small p-value provides strong statistical evidence to reject the null hypothesis (H_0), which posits no mean difference between the pre-test and post-test scores. These results support the research hypothesis that the game-based platform has a significant impact on improving student learning outcomes. The statistically significant mean difference underscores the platform's success in enhancing students' understanding and skills in programming learning. These findings not only validate the effectiveness of the intervention but also demonstrate that the platform can serve as an innovative, evidence-based learning tool to support the enhancement of learning quality in the future.

Fig. 4 illustrates the distribution of final scores, which are overall significantly higher than the initial scores, with no substantial overlap between the quartiles of the two groups. This visualization provides strong evidence of a clear difference between the initial and final scores, indicating a significant improvement in students' skills after using the game-based platform. Meanwhile, Fig. 5 shows the distribution of score improvements, with the majority of students exhibiting a positive shift. This demonstrates that the platform effectively enhanced programming skills across students, even though there was some variation in the degree of improvement among individuals. Such variation suggests that, while the platform's overall effectiveness is robust, its impact may differ based on factors such as students' initial abilities or how they utilized the platform's features. Overall, these findings confirm the success of the game-based platform in supporting effective learning and substantially improving students' programming skills.

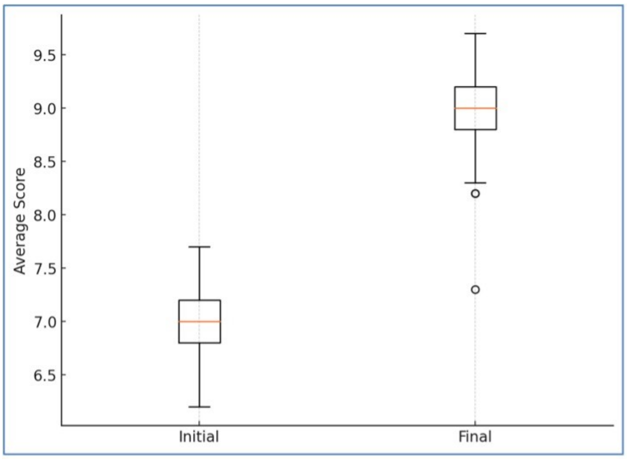

Fig. 4. Comparison of initial and final scores.

Further analysis of item-level responses revealed that three game elements contributed most significantly to improved engagement and learning outcomes. First, the live coding feature with real-time feedback allowed students to receive immediate, actionable responses to their input, which helped reduce frustration and reinforce learning. Second, the use of progressively structured challenges encouraged students to build their skills incrementally, fostering a sense of achievement. Third, the leaderboard and badge reward system created a motivational environment through healthy competition, which was especially effective for students with strong self-regulated learning tendencies. These elements worked together to promote both intrinsic and extrinsic motivation within the learning process.

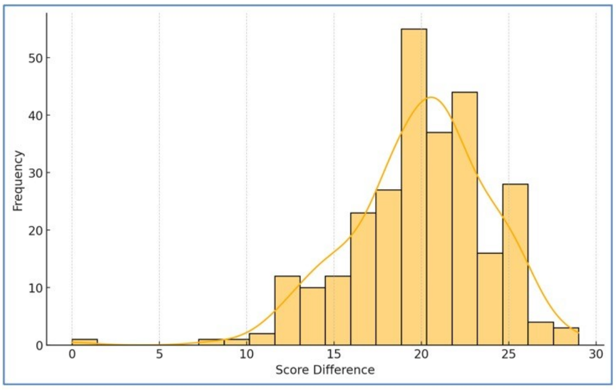

Fig. 5. Distribution of difference scores.

Thus, the significant difference between the initial and final scores underscores the tangible impact of the game-based platform in enhancing students' programming abilities. This innovative learning intervention effectively improved learning outcomes significantly, both statistically and practically. The average increase of nearly 20 points reflects a substantial change, indicating that the platform not only meets learning needs but also assists students in approaching the expected maximum performance levels. These results highlight the platform's effectiveness in supporting the achievement of optimal learning objectives while positively contributing to the development of students' programming skills. These findings reinforce the relevance and potential of game-based platforms as innovative learning tools to support academic success and prepare students for the challenges of increasingly complex technological environments.

In addition, system logs captured in-platform data such as total XP, time spent per challenge, number of attempts, and badge completion. While these platform metrics were not used in statistical tests in this version of the study, they informed formative platform evaluation and may be explored in future analytics-based research.

V. Conclusion

This study demonstrates that the GBL platform significantly enhances student engagement and learning outcomes in programming learning. Descriptive and inferential statistical analyses revealed substantial improvements in engagement and learning scores following the platform's implementation. The average engagement score increase of 208.62 points and the learning outcome improvement of 19.94 points indicate a significant positive impact, both statistically and practically, on students' learning experiences.

These results confirm both research hypotheses. H_1, regarding a significant increase in student engagement, and H_2, concerning a significant improvement in programming learning outcomes, are both supported by the empirical data. This confirmation reinforces the platform's effectiveness and the validity of the study's experimental design.

These findings reinforce the core assertion that the GBL approach, designed with interactive elements such as live coding, leaderboards, and progressive challenges, effectively motivates students to actively participate and enhance their programming skills. The platform not only aids students in gaining a deeper understanding of programming concepts but also facilitates an enjoyable and meaningful experiential learning process.

The implications of this research are highly relevant to higher education, particularly in STEM fields, where student engagement and active participation present significant challenges. The findings indicate that the integration of game elements can serve as an innovative solution to enhance the quality of learning and drive improved academic achievement.

Despite the positive outcomes, this study has several limitations. First, it employed a single-group pre-test/post-test design without a control group. This limits the ability to attribute causality solely to the intervention, as other contextual or instructional variables may have influenced the outcomes. Future studies should include a randomized control or comparison group to better isolate the effects of the GBL platform. Second, the findings are based on students from a single university, which may affect the generalizability of the results. Third, while platform usage data were collected, this version of the study primarily relied on quantitative measures and did not fully incorporate interaction logs or deeper behavioral analytics.

Future studies should explore the application of the platform across diverse academic settings, programming languages (e.g., Python, Java), and learner populations (e.g., beginners vs. advanced coders). Longitudinal studies could be conducted to examine the impact of game-based platforms on long-term skill retention and career-readiness. Additionally, future research should leverage platform log data to conduct learning analytics, enabling deeper insights into student behavior, time-on-task, and adaptive learning patterns. It is also recommended to explore the integration of artificial intelligence to provide personalized feedback and adaptive challenge levels based on learner performance in real-time.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Noven Indra Prasetya conducted the research, wrote the paper, and analyzed the data; Nia Saurina proofread and finalized the article. All authors had agreed and approved the final version.

Funding

This research was funded by Universitas Wijaya Kusuma Surabaya, grant number SK Rektor Nomor 63 Tahun 2025.

Acknowledgment

The authors wish to thank Universitas Wijaya Kusuma Surabaya.

References

[1] D. Topalli and N. E. Cagiltay, "Improving programming skills in engineering education through problem-based game projects with Scratch," *Comput. Educ.*, vol. 120, pp. 64–74, May 2018.

[2] B. Wahono *et al.*, "Development of integrated STEM education learning units to access students' systems thinking abilities," *J. Inov. Teknol. Pendidik.*, vol. 10, no. 1, pp. 1–9, Mar. 2023.
[3] J. Zhang and Z. Hu, "Advancing game-based learning in higher education through debriefing: Social constructivism theory," *J. Educ. Gift. Young Sci.*, vol. 12, no. 1, pp. 15–27, 2024.
[4] S. A. Ishak, R. Din, and U. A. Hasran, "Defining digital game-based learning for science, technology, engineering, and mathematics: A new perspective on design and developmental research," *J. Med. Internet Res.*, vol. 23, no. 2, e20537, 2021.
[5] A. S. Munna, D. Ilham, and K. Sandi, "Bridging the gap: Effective training methods for enhancing STEM skills in higher education," *J. Train. Educ. Sci. Technol.*, vol. 1, no. 1, pp. 7–12, Dec. 2023.
[6] J. Ilić, M. Ivanović, and A. Klašnja-Milićević, "Effects of digital game-based learning in stem education on students' motivation: A systematic literature review," *J. Balt. Sci. Educ.*, vol. 23, no. 1, pp. 20–36, 2024.
[7] S. Subiyantoro *et al.*, "Developing gamified learning management systems to increase student engagement in online learning environments," *Int. J. Inf. Educ. Technol.*, vol. 14, no. 1, pp. 26–33, 2024.
[8] N. I. Prasetya, S. Syidada, and L. Retnawati, "Development of serious games as a programming learning platform for informatics students," *J. Paedagogy*, vol. 12, no. 1, pp. 12–22, Jan. 2025.
[9] H. Y. Lee *et al.*, "Exploring the learning process and effectiveness of STEM education via learning behavior analysis and the interactive-constructive-active-passive framework," *J. Educ. Comput. Res.*, vol. 61, no. 5, pp. 951–976, 2023.
[10] N. A. A. Zaki *et al.*, "Developing a conceptual model of learning analytics in serious games for STEM education," *J. Pendidik. IPA Indones.*, vol. 9, no. 3, pp. 330–339, 2020.
[11] I. N. S. Degeng, "The effect of PBL-based STEAM approach on the cognitive and affective learning outcomesof primary school," *Turk. J. Comput. Math. Educ.*, vol. 12, no. 6, pp. 2390–2399, 2021.
[12] S. Patmanthara *et al.*, "The use of ladder snake games to improve learning outcomes in computer networking," *Int. J. Emerg. Technol. Learn.*, vol. 14, no. 21, pp. 243–249, 2019.
[13] E. V. Soboleva *et al.*, "Applying gamification in learning the basics of algorithmization and programming to improve the quality of students' educational results," *Eur. J. Contemp. Educ.*, vol. 10, no. 4, pp. 987–1002, 2021. (in Russian)
[14] S. A. Thorat and D. P. Kshirsagar, "Developing logic building, problem solving, and debugging programming skills among students," *J. Eng. Educ. Transform.*, vol. 34, pp. 402–406, Jan. 2021.
[15] N. Akmar *et al.*, "Design and development EduPocket A+," *Momentum Phys. Educ. J.*, vol. 5, no. 1, pp. 94–100, 2021.
[16] C. Johnson, I. Govender, and D. W. Govender, "Digital game-based learning for information technology: An exploratory analysis," *E-J. Humanit. Arts Soc. Sci.*, vol. 3, no. 13, pp. 730–744, Dec. 2022.
[17] S. Ulfa, D. B. Lasfeto, and I. Fatawi, "Applying fuzzy logic to customize learning materials in e-learning systems," *Ubiquitous Learn. Int. J.*, vol. 14, no. 2, pp. 49–61, 2021.
[18] H. Li and J. P. G. Cheong, "Using the ADDIE model to design and develop physical education lessons incorporated with a functional training component," *Front. Public Health*, vol. 11, 1201228, 2023.
[19] S. Campbell *et al.*, "Purposive sampling: Complex or simple? Research case examples," *J. Res. Nurs.*, vol. 25, no. 8, pp. 652–661, 2020.
[20] Purwanto and P. B. D. Hedin, "Measurement of user satisfaction for web-base academic information system using end-user computing satisfaction method," in *Proc. IOP Conf. Ser. Mater. Sci. Eng.*, 2020, 012044.
[21] W. Waskito *et al.*, "Constructivist feedback-based assessment method as key for effective teaching and learning: The development and impact on mechanical engineering students' adaptive capacity, decision making, problem solving and creativity skills," *Int. J. Cogn. Res. Sci. Eng. Educ.*, vol. 12, no. 1, pp. 57–76, 2024.
[22] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions," *Contemp. Educ. Psychol.*, vol. 61, 101860, Apr. 2020.
[23] D. Samarasinghe, M. Barlow, and E. Lakshika, "Flow-based reinforcement learning," *IEEE Access*, vol. 10, pp. 102247–102265, 2022.
[24] F. Pradana *et al.*, "Development of gamification-based e-learning on web design topic," *Int. J. Interact. Mob. Technol.*, vol. 17, no. 3, 21, Feb. 2023.
[25] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, New York: Routledge, 2013, ch. 2.