Effectiveness of Gamification Elements in Teaching Programming to Future Informatics Teachers

Aigul Aldabergenova[1], Ulzhalgas Yesseikyzy[1,*], Ilyas Yessengabylov[1], Aiym Yesseikyzy[1], and Assem Yerkinova[2]

[1]Department of Informatics and Digitalization of Education, Faculty of Physics and Mathematics,
Zhetysu University named after I. Zhansugurov, Taldykorgan, Kazakhstan
[2]Non-Commercial Educational Association Swiss Innovators Club, Leysin, Switzerland
Email: aigul_ao@mail.ru (A.A.); yesseikyzy@gmail.com (U.Y.); ilias_e@mail.ru (I.Y.); aiymyesseykyzyy@gmail.com (A.Y.); assem.yerkinova@swissinnovatorsclub.ch (A.Y.)
*Corresponding author

Manuscript received August 13, 2025; revised September 9, 2025; accepted September 18, 2025; published March 10, 2026

***Abstract*—This study investigates the effectiveness of gamification elements in teaching programming to future Informatics teachers, using Python as a case study. Gamification incorporates game mechanics such as points, badges, leaderboards, and narratives into educational settings to enhance student engagement, motivation, and academic performance. A quasi-experimental design was employed, comparing outcomes of an experimental group engaged in gamified programming tasks with those of a control group following traditional learning methods. Independent-samples t-tests were used to assess learning gains. Results revealed that the experimental group achieved scores between 85–87 points, significantly outperforming the control group, which averaged 63–65 points. These findings suggest that gamification was associated with greater gains in coding, problem-solving, and algorithmic thinking skills among students while addressing challenges like diminished motivation in conventional learning environments. Moreover, the study indicates that structured gamification may be aligned with curriculum goals to promote deeper cognitive engagement. Future research may examine how adaptive or artificial intelligence-enhanced gamification tools could expand these benefits across diverse learner populations.**

***Keywords*—gamification, programming, game mechanics, informatics education, academic performance**

I. Introduction

Gamification refers to the learning process where students solve problems and overcome challenges in game-based environments to achieve core learning outcomes, which is incorporated in either badges, points, and leaderboards or complex forms of virtual simulation environments [1]. Similarly, Bodnar *et al*. [2] posit that gamification is the use of game elements and game mechanics in a non-context that allows achieving goals and assessing the level of success. Gamification tends to positively impact students' academic performance and self-efficacy, as well as motivate learners to engage in scientific inquiry activities [3]. Gamification also acts as a positive element in learning because it enhances students' positive sentiments once they complete tasks successfully; conversely, it generates anxiety when they fail tasks. Domínguez *et al*. [4] assert that gamification has been successfully integrated into platforms, particularly social ones, as a method to foster close connections between the platform and its users, promoting viral behaviours that boost platform popularity. Nonetheless, gamification has been recognised as an essential element in improving online education, with the ability to tackle student challenges like diminished motivation resulting from insufficient interaction with instructors and peers. Fotaris and Mastoras [5] showed that gamified approaches can enhance student engagement and improve learning outcomes in computing education. Gamification is also applied in various sectors such as marketing, to engage customers and inspire employees.

In education, various types of games are employed in learning, including Game-Based Learning (GBL). Zainuddin *et al*. [6] note that specific learning objectives can be embedded into gameplay and game mechanics systematically integrated into instruction, thereby aligning learning with engaging activities.

According to Ermakan *et al* [7], there are two primary gamification categories: structural and content gamification. Structural gamification refers to the use of game features to engage learners with the material without altering its content, utilizing features like points, levels, badges, leaderboards, and achievements and incorporating them into an educational setting. Content gamification, conversely, involves using game elements, mechanics, and thought processes to transform the content, making it resemble a game more closely. In addition to game features, an alteration to the content exists by incorporating a narrative, challenges, intrigue, mystery, and characters to captivate the learner.

Similarly, one interesting strategy in programming languages is gamification [8]. Essentially, gamification in computer programming courses extends beyond mere entertainment by integrating gamification components to achieve educational objectives and fostering an encouraging atmosphere for students to learn. The significance of gamification relies on the use of game mechanics such as points, badges, levels, and leaderboards to establish a competition-oriented atmosphere that guides learners' actions and encourages them to learn [9].

Education gamification implementation strategies are considered to be essential for such educational courses that demand high cognitive involvement such as computer programming languages. Therefore, education gamification is considered an effective method of enhancing student motivation and interest in performing, enabling them to learn how to solve problems. This is because gamification adds elements of excitement, interest, and fun, which result in higher student engagement and motivation levels [10]. It is argued that the incorporation of games into education can enhance students' motivation and involvement in learning, resulting in the acquisition of long-term practical

doi: 10.18178/ijiet.2026.16.3.2532

knowledge [11]. The reason is that gamification helps to stimulate learning environments, which is crucial in fostering students' critical thinking, problem-solving skills, creativity, and collaboration, thus equipping them for life in a rapidly evolving world [12].

Consequently, it is perceived that the creation of engaging and interactive games, mainly for educational purposes, is considered progress in applying game-based teaching. Thoughtful games allow players to enhance their understanding and refine their abilities by navigating various gaming hurdles [13]. Also, game-based learning enables students to learn at their own pace and in a manner that is aligned with their personal needs and learning styles, as they can ask questions about various situations, make decisions, and understand the outcome of their actions without fear of errors.

The game-based learning approach to computer programming training, according to Kaldarova *et al.* [14], enables students to acquire complicated principles and skills in an entertaining and interactive manner, with the formation of critical thinking and teamwork while they learn algorithms thinking, data structures, networks, software testing, and programming languages through the design and evaluation of their games and simulations. In addition, gamification is said to help bridge the theory-practice gap as it enables students to implement knowledge in practical situations. Tsur and Rusk [15] argue that students can create animations, interactive stories, and games, enabling them to code, learn programming concepts, and even learn other topics in computer science through game creation. The argument is that programming courses are the most prevalent in computer science, yet studying programming is viewed as a significant barrier. For example, studying object-oriented programming languages is a substantial obstacle for students due to the cognitive demands of programming, the need for algorithmic thinking and problem-solving ability, and the reality that it is a lengthy process.

Subhash and Cudney [16] noted in their systematic review that gamification, when used as a teaching or learning method in computer programming can encourage learners to respond quickly and think innovatively. Therefore, this implies that students largely engage in playing games not just for playing's sake but because they wish to enhance their know-how or competency in the computer programming language that Informatics instructors teach. It is also theorized that gamification may influence student behaviors, for instance, by making them feel like they are superheroes playing a game, hence prompting them to assist other students in overcoming problems within the game, thus triggering student-centered learning. To this end, gamification has, over time, gained recognition as a novel educational strategy that actively involves students in their learning experiences.

By successfully integrating game components, students are likely to show enhanced usability, educational effectiveness, and an improved overall experience during their learning journey [17]. The incorporation of gamification elements in teaching computer programming students has been found to encourage learners to participate for longer durations, resulting in more effective learning while also improving their critical thinking, problem-solving, and decision-making abilities [18].

II. Literature Review

A. The Effectiveness of a Gamified Programming Education

Gamification has been a widely discussed issue for enhancing instruction and learning in computer programming in which abstract concepts and steep learning curves discourage students. Landers [19] proposed a theoretical distinction between gamification and game-based learning in which gamification is presented as an intermediate entity between instructional design and learning. While this model is theoretically illuminating, its lack of empirical validation makes it difficult to implement in diverse learning settings, particularly in computer programming, where motivation and resolve are key to success.

Smiderle *et al.* [20] extended this argument by illustrating that students' personality can have a considerable influence on how much they adjust to gamified coding environments. Their study, with its caveat of a limited sample size ($n = 40$), highlights that gamification is not a "one-size-fits-all" strategy. Similarly, Codish and Ravid [21] highlight that learner traits govern interaction with gamified systems and that in the absence of a thoughtful regard for diversity among students, then gamification will tend to worsen not reduce inequality. This paper emphasizes the need to design interventions with much care to consider differences in learners' prior knowledge, digital literacy, and motivational profiles.

Despite these cautions, there are many encouraging consequences of gamification in coding instruction reported in different studies. Gamified programming courses using C++, C#, and Java, for example, have been associated with greater motivation and comprehension [22]. Swacha and Szydłowska [23] also argue that programming is most appropriate for gamification due to the sophistication of the topic as well as the existence of automated feedback, which is able to provide instant reinforcement as well as correcting mistakes. Their enthusiasm overlooks potential traps such as overreliance on external rewards or reduced internal involvement in instances of poorly crafted game mechanisms.

Quasi-experimental studies also provide evidence of significant learning gains. Marín *et al.* [24] reported statistically significant differences in the performance of students when they learned C programming using a gamification platform. Pradana *et al.* [25] also reported large differences between treatment effects of gamified and non-gamified learning outcomes. These findings suggest that gamification can yield more effective and motivating programming environments. Methodological problems, however, deflate the power of these findings. Both tests are susceptible to confounding novelty with actual pedagogical power, since gains in performance in the short term may not generalize to strong learning. In addition, they mainly label higher grades as better learning, a poor measure that overlooks richer metacognitive, cognitive, and affective dimensions of programming mastery.

A second significant issue is the reported result clarity. Some articles report "outstanding scores" or general improvements without reporting particular benchmarks, statistical information, or standardized tests. This looseness restricts the comparability of results across research and degrades the evidence-based argument of gamification as a programming educational approach. As Király and Balla [22] and Durmaz *et al*. [26] later comment on in their broader reviews, gamification research in programming typically has small samples, low-quality designs, and unreliable reporting, so its true impact remains to be seen.

Overall, while literature is generally favorable towards gamification in computer programming, it also has severe gaps. The highly reported positive results may be a result of inflated novelty effects, selective reporting, or short-term outcome measures. Likewise, the heterogeneity of learners' reactions based on personality, gender, and prior experience disconfirms the belief that gamification can be universally effective. The general evidence suggests that gamification holds potential for computer programming teaching but that its real value lies not in blanket adoption, but in carefully designed interventions for specific learners and environments.

B. The Influence of Game Mechanics on Student Motivation and Academic Performance in Programming Courses by Computer Programming Teachers

Gamification studies in computer science education are increasingly examining the impact of specific mechanics such as points, badges, leaderboards, or collaboration on motivation and learning performance. With consensus that the mechanics can engage students better, evidence remains mixed regarding their sustainability, pedagogical depth, and potential unintended side effects.

Adam and Berg [27] illustrate this argument by illustrating that gamified online learning quizzes enhanced students' motivation, understanding, and self-efficacy. But the studies also concluded that these quizzes were most effective in low-stakes environments. This raises issues of transferability: whereas quizzes can enhance participation where the risk of failure is minimal, it is not so clear that they are useful in high-stakes or complex programming tasks. The assumption that mechanics in a natural way tend to cause high engagement risks oversimplifying the phenomenon of student learning, influenced as it is by prior knowledge, learning styles, and learning stakes of the environment.

Durmaz *et al*. [26] provide more solid evidence that reward mechanics, points, badges, and leaderboards are capable of fostering success, power, and affiliation motives. Their findings, however, point to the general criticism of gamification research: the excessive focus on a narrow range of extrinsic motivators. Whereas effective in the short-term for engaging people, such mechanics are actually counterproductive when overly applied to intrinsic motivation, echoing larger problems in motivational psychology about the "over justification effect". Such reliance also limits experimentation with other mechanics, such as narrative or exploratory ones, that can result in more intense interaction with programming material.

User experience and usability are also common issues. Azzali *et al*. [17] report high satisfaction with gamified computer programming courses, highlighting the role of usability in forming learners' effectiveness beliefs. But their conclusion is heavily reliant on self-reports and therefore could be biased. More critically, it does not take into account that increased usability automatically results in increased learning outcomes without examining whether gamification actually strengthens higher-order learning abilities such as critical thinking or problem-solving. This inconsistency also reflects a broader failing in the literature: many studies privilege surface-level indicators of engagement over more insightful cognitive results.

Alternative studies locate game mechanics within broader theoretical frameworks. For instance, Proulx *et al*. [28], using Self-Determination Theory (SDT), argue that social discovery mechanics can foster autonomy and self-determination. Though such research shows potential for gamification to increase intrinsic motivation, many issues remain unresolved. It is not clear whether increased motivation manifests as long-term skill development, and possible drawbacks such as social comparison or competitive pressure may fuel anxiety more than support learning.

Evidence from programming environments adds further complexity. Jusas *et al*. [29] suggest that elements such as experience points, engaging content, and cooperation support sustainable learning behaviors in Object-Oriented Programming (OOP) courses. However, their study does not clarify whether these behaviors result from genuine cognitive improvements or the novelty of gamified activities. Similarly, Vlachopoulos and Makri [30] identify feedback as a key gamification feature in higher education. While they promote understanding the importance of transferring knowledge across contexts, their qualitative research is too weak to establish causal links between specific mechanics and measurable academic achievements.

In a nutshell, this literature is insistent that although game mechanics can actually facilitate learning, their benefit is short-term, context-dependent, and liable to negative side effects. Extrinsic rewards like points and badges, if used excessively, can undermine intrinsic interest, while usability and self-report measures hide whether more significant cognitive effects are attained. At the same time, promotion of SDT-based theory development suggests that those mechanics supporting autonomy, competence, and relatedness can offer a means for more stable learning. What are less well examined, though, is how these mechanics work together in varying educational settings and how their influence changes over time.

C. The Impact of Gamification on the Advancement of Coding Skills, Problem-Solving Abilities, and Algorithmic Thinking

More and more research has investigated the impact of gamification on higher-order cognitive products such as coding abilities, problem-solving potential, and algorithmic thinking. While there is some evidence that gamification can support these capabilities, the results are so far piecemeal regarding generalizability, long-term performance, and side effects. In this regard, Kaya and Ercag [31] developed a challenge-based gamification software using self-determination theory and flow theory and reported the experimental group to be more motivated, confident, and problem-solving. Despite being theory-based, the limited sample size of the study (n = 30 each) means it cannot be generalized, and the short-term-only study does not say anything about whether gains are sustained improvements. Moreover, it does not address the question of whether gamification is inducing extrinsic motivation, which could undermine intrinsic interest in the long run.

Kaya and Ercag [31] broaden the discussion by recognizing both the positive and the negative impacts of gamification. They identify strengths of increased logical thinking and problem-solving, but also warn of threats like addiction and decreased social interaction. While this two-dimensional perspective is useful, the absence of empirical evidence on negative impacts negates their conclusions. Their suggestion to have learners monitor progress and reflect on improvement tilts towards self-directed learning but possibly towards being preoccupied with reward to the detriment of deeper engagement with underlying programming concepts.

Setting and age also appear to affect gamification's impact on cognition. Sun and Liu [32] found that gamified Python programming enhanced sixth-grade students' computational thinking with significant gender differences: boys excelled in pattern recognition, while girls excelled in abstraction and decomposition. These findings suggest promise for gamification to enhance domain-specific cognitive skills but fail to explore the social and experiential explanations for such differences. Without such analysis, the results are likely to perpetuate gender stereotypes rather than be employed to inform inclusive instructional design.

Additional research shows that gamification may elevate motivation but not necessarily improve cognitive attainment. Chen *et al*. [33], for instance, used a puzzle game in teaching programming concepts and found high levels of flow and engagement but no difference in performance between gamified and non-gamified groups. Similarly, Kaldarova *et al*. [14] reported that their game-based intervention in primary school enhanced students' motivation and interest in computer science, yet the study design did not account for novelty effects or social influence. These instances draw attention to a shared failing: boosts in engagement are often assumed to constitute evidence of cognitive gains, without explicit demonstration of long-term skill building.

Hamari *et al.* [34] observed that the motivational impact of gamification can vary across learners, with some students responding strongly to elements such as digital rewards while others remain less influenced. Their findings also indicated that traditional assessment methods, such as grades, continue to serve as the strongest motivators in higher education, suggesting clear limits to the extent to which gamification can replace established measures of evaluation. In this regard, Czakóová [35] illustrates the strength of fun-based problem-solving approaches in cultivating computational thinking, but provides poor empirical evidence for linking gamification with cognitive gain that can be measured directly.

Together, these studies offer a varied and context-dependent result. While gamification can stimulate motivation and, on occasion, support problem-solving and computational thinking, evidence of long-term cognitive benefits is limited. Many studies rely on short or specialized samples, short interventions, or measures of participation that do not reflect learning directly. So too do gender differences, differential motivational responses, and potential negative side effects like addiction or external dependence complicate unjustified effectiveness claims. Future research will have to sort out short-term motivational improvements from long-term learning improvements, employ stronger measures of cognitive skill acquisition, and test gamification across multiple levels of education and learner populations.

III. Materials and Methods

A. Research Questions

The purpose of this study is to assess the effectiveness of gamification in a programming course (using Python as an example) and its impact on the academic achievements and skills of future Informatics teachers. The research questions are:

1) How does gamification affect student engagement, motivation, and understanding in programming courses, specifically regarding coding, problem-solving, and algorithmic thinking?
2) What is the impact of gamified learning platforms on student academic performance compared to traditional teaching methods?
3) How do game mechanics (e.g., rewards, achievements, levels) influence student motivation and skill development in programming courses?

To address these questions, the research was divided into two parts: Study A and Study B. Study A focused on a survey investigating motivation, preferences, and perceptions of gamification, while Study B was a quasi-experiment testing the effect of gamification on programming outcomes.

B. Study A: Survey on Perceptions and Motivation

1) Research participants

The study involved 203 participants enrolled in the educational programs "Informatics" (6B01505) and "Mathematics-Informatics" (6B01502) at the following universities:

Zhetysu University named after I. Zhansugurov,

Kazakh National Women's Pedagogical University,

Caspian University of Technology and Engineering named after Sh. Yessenov.

The participants in Study A were selected using purposive sampling to capture a broad representation of students, including but not limited to those in the quasi-experiment of Study B.

2) Data collection methods (Study A)

A survey was conducted to assess student preferences, motivation, and perceived effectiveness of gamification in education. The survey contained five questions, with multiple-choice responses, and was specifically designed to evaluate how students perceived gamification in their learning process.

3) Data analysis (Study A)

The quantitative data from the survey were analysed using descriptive statistics, such as mean scores and distribution patterns. This analysis provided insight into how students viewed the role of gamification in enhancing motivation, engagement, and skill development.

Participation was voluntary and informed consent was obtained (see Ethics and Consent).

C. Study B: Quasi-Experiment on Learning Outcomes

1) Research participants (Study B)

The study involved 90 first-year students enrolled in the educational programs "Informatics" (6B01505) and "Mathematics-Informatics" (6B01502) at the same three universities:

- Zhetysu University named after I. Zhansugurov (30 students),
- Kazakh National Women's Pedagogical University (30 students),
- Caspian University of Technology and Engineering named after Sh. Yessenov (30 students).

The sample was divided into two groups:

- Experimental group (*n* = 45): students who used Blockland.kz, a gamified educational platform developed by the authors of the study.
- Control group (*n* = 45): students who received instruction through traditional methods (lectures, seminars, and practical exercises without gamification elements).

The participants were selected using purposive sampling to ensure similar baseline knowledge levels. Study B spanned fifteen weeks and covered fundamental Python programming concepts. The learning objectives included coding proficiency, problem-solving abilities, algorithmic thinking and computational thinking. The skills assessed included coding, problem-solving, and algorithmic thinking.

2) Data collection methods (Study B)

A quasi-experiment with Pre-test and Post-test assessments of students' programming skills was carried out:

- Pre-test: Assessed baseline knowledge in syntax, conditional statements, and loops. Both groups performed similarly (currently "50–60" points out of 100).
- Post-test: Evaluated advanced skills through coding tasks, code analysis, logic analysis, and debugging. Coding, problem-solving, and algorithmic thinking were assessed using a structured instrument developed for this study. Each domain was evaluated on a 100-point scale, with tasks aligned to the learning objectives of the programming course. The instrument demonstrated high internal consistency, with Cronbach's $\alpha = 0.89$, confirming the reliability of the assessment.
- Coding skills were measured through tasks requiring students to write executable Python code to solve specific problems, such as loops, conditionals, and data handling.
- Problem-solving skills were assessed using scenario-based questions where students identified logical solutions and explained their reasoning.
- Algorithmic thinking was evaluated through structured tasks requiring stepwise planning, decomposition of problems, and pseudocode/flowchart construction.

The control group improved to 60–70 points, while the experimental group achieved 80–90 points. The results were analysed to compare the learning progress between the experimental and control groups.

3) Data analysis (Study B)

The quantitative data from the quasi-experiment were analysed using IBM SPSS Statistics (Version 28). Descriptive statistics (means and standard deviations) were first calculated for pre-test and post-test scores. To assess the effectiveness of gamification, independent-samples t-tests on gain scores were conducted to compare improvements between the control and experimental groups. Prior to hypothesis testing, assumption checks were performed: the Shapiro–Wilk test and Q–Q plots indicated approximate normal distributions, and Levene's Test confirmed homogeneity of variances ($p > 0.05$). As a robustness check, Welch's correction was also applied, yielding consistent results. In addition to p-values, effect sizes (Cohen's d) and 95% confidence intervals were reported to evaluate the magnitude and precision of group differences.

This study was approved by the Ethics Committee of Zhetysu University (Approval No. 1473, dated 12/06/2025). All participants provided written informed consent before data collection. Anonymity and confidentiality were maintained throughout the research, and participants had the right to withdraw at any stage without consequences.

IV. Results

A. Study A: Survey on Perceptions and Motivation (n = 203)

1) Demographic data

The survey data are derived from self-reports of 203 future teachers enrolled in "Informatics" and "Mathematics-Informatics" educational programs. The respondents represented three universities: Zhetysu University named after I. Zhansugurov (47.3%), Kazakh National Women's Pedagogical University (28.1%), and Caspian University of Technology and Engineering named after Sh. Yessenov (24.6%) (Table 1).

Table 1. University affiliation

University	Responses	Percentage (%)
Zhetysu University named after I. Zhansugurov	96	47.3
Kazakh National Women's Pedagogical University	57	28.1
Caspian University of Technology and Engineering named after Sh. Yessenov	50	24.6
Total	203	100

Furthermore, based on the respondents' specialization, the results showed that 21.2% were Mathematics-Informatics students, whereas 78.8% were Informatics majors, indicating the predominance of Informatics students (Table 2).

Table 2. Specialization

Specialization	Responses	Percentage (%)
Mathematics-Informatics	43	21.2
Informatics	160	78.8
Total	203	100

Concerning gender, the results showed that female students were the majority, with 67.5% compared to males 32.5%, thus implying that there is a high enrolment of females in Informatics and Mathematics-Informatics courses (Table 3).

Table 3. Gender

Gender	Responses	Percentage (%)
Male	66	32.5
Female	137	67.5
Total	203	100

2) Preferred formats and motivational elements

Table 4 presents information about the most effective and convenient format of material delivery for understanding programming courses. According to self-reports, 37.9% (95% CI [31.2, 45.0]) of the students selected gamified elements as the most effective and convenient format, followed by interactive tasks and project-based learning with 29.1%, while traditional lectures accounted for 23.2%. Independent study was least preferred at 9.8% (95% CI [6.2, 14.8]).

Table 4. Format of material delivery found most effective and convenient

Format of material delivery	Responses	Percentage (%)	95% CI
Traditional lectures and seminars	47	23.2	[17.7, 29.8]
Interactive tasks and project-based learning	59	29.1	[23.0, 36.1]
Gamified elements (quests, levels, rankings, rewards)	77	37.9	[31.2, 45.0]
Independent study of theory	20	9.8	[6.2, 14.8]

Regarding learning elements that increase motivation, 43.8% (95% CI [36.9, 50.9]) of the respondents selected rewards and achievements for completing assignments as the most important element, followed by competition with classmates at 24.6% (95% CI [18.8, 31.5]) and gamified tasks with different difficulty levels at 20.2% (95% CI [14.9, 26.7]). Only 11.4% (95% CI [7.6, 16.5]) reported no preference for gaming elements (Table 5).

Table 5. Learning elements and motivation

Learning elements that increase motivation	Responses	Percentage (%)	95% CI
Rewards and achievements for completing assessments	89	43.8	[36.9, 50.9]
Compete with classmates	50	24.6	[18.8, 31.5]
Gamified tasks with different difficulty levels	41	20.2	[14.9, 26.7]
No gaming elements–I feel comfortable without them	23	11.4	[7.6, 16.5]
Total	203	100	-

3) Perceived effects of game mechanics

Analysis of the responses showed that rewards and achievements for completing assignments were considered the most powerful motivational factor (43.8%), followed by collaboration with classmates (24.6%). Gamified tasks with different levels of difficulty ranked third (20.2%), suggesting a comparatively weaker perceived influence on motivation. For clarity, key proportions with 95% CIs are: rewards/achievements 43.8% [36.9, 50.9], peer collaboration 24.6% [18.8, 31.5], varying difficulty 20.2% [14.9, 26.7], and "no gaming elements" 11.4% [7.6, 16.5].

The results presented in Table 6 above show that a majority of the respondents agreed that game mechanics such as achievements, levels, competitions, and game-based tasks significantly increased their motivation and academic performance (54.2% [47.1, 61.1]), whereas 32.0% reported a slight increase and 13.8% indicated no change. Fig. 1 represent perceived effects (self-reports) rather than objective performance outcomes, and 95% CIs are provided to indicate the precision of the estimates.

Table 6. Game mechanics and their effect on student motivation and academic performance

Game Mechanics (achievements, levels, competitions, game-based tasks)	Responses	Percentage (%)	95% CI
Significantly increase motivation and performance	110	54.2	[47.1, 61.1]
Slightly increase motivation	65	32.0	[25.8, 38.9]
Would not change anything	28	13.8	[9.5, 19.5]
Total	203	100	

The results presented in Table 7 show that a majority of the students self-reported that the use of game mechanics in programming courses significantly increased their engagement (61.1% [54.0, 67.8]), whereas 23.6% indicated only a slight increase in interest and 15.3% reported no change. This distribution suggests that while most students perceived gamification as beneficial, a notable minority did not view it as influential.

Table 7. Game mechanics in programming courses effect on student engagement

Effect of game mechanics in programming on student engagement.	Responses	Percentage (%)	95% CI
Significantly increase motivation	124	61.1	[54.0, 67.8]
Slightly increase interest	48	23.6	[18.0, 30.2]
Would not change anything	31	15.3	[10.8, 21.2]
Total	203	100	

Furthermore, when asked to what extent game mechanics enhanced their learning interest, respondents gave an average rating of 4.3/5, reflecting a generally positive assessment of gamification. These findings imply that game elements can make programming courses more engaging and motivational, although the benefits are not uniform across all learners.

B. Study B: Quasi-Experiment on Learning Outcomes (n = 90)

1) Baseline equivalence

The quasi-experiment took 15 weeks, which entailed an intensive block on Python programming fundamentals. The evaluated skills were based on coding, problem-solving, and algorithmic thinking. Therefore, the methods used to assess these skills were two (pre-test and post-test). At the pre-test, basic knowledge of syntax, conditional statements, and loops was assessed, and as such, both groups (experimental and control group) showed comparable results (50–60 points out of 100, see Table 8).

In the post-test assessment method, complex tasks such as writing a small program, analysing code logic, and identifying errors were conducted in both groups.

Nevertheless, unlike the pre-test stage, the control groups showed improved performance scores of 60–70 points, while the experimental group had an improved results score of 80–90, indicating more substantial progress.

Table 8. Baseline equivalence of pre-test scores (Welch’s t-tests, df ≈ 87–88)

Skill Domain	Control Group (*n* = 45) (M ± SD)	Experimental Group (*n* = 45) (M ± SD)	Mean Diff (Exp – Ctrl)	*t* (df)	*p*	Cohen’s d	95% CI
Coding	56.2± 2.8	56.4 ± 2.9	0.2	–0.33 (88)	0.74	0.07	[–0.99, 1.39]
Problem-Solving	55.4 ± 2.9	57.0 ± 3.1	1.6	–2.53 (88)	0.013	0.53	[0.34, 2.86]
Algorithmic Thinking	57.0 ± 3.0	56.5 ± 2.7	–0.5	0.83 (87)	0.41	–0.18	[–1.70, 0.70]

Independent-samples Welch’s t-tests were conducted to assess baseline equivalence between groups. Degrees of freedom vary slightly across tests due to Welch’s adjustment.

Although baseline equivalence was observed for coding and algorithmic thinking, the problem-solving domain showed a statistically significant difference at pre-test ($p = 0.013$). To address this, an ANCOVA was conducted with pre-test scores entered as covariates. The analysis confirmed that the experimental group still outperformed the control group in all three domains, with adjusted post-test means showing significant differences in coding, problem-solving, and algorithmic thinking (all $p < 0.001$). These findings indicate that the observed effects were robust even after controlling for baseline differences.

2) Skill assessment results

The skill assessment was conducted using a 100-point scale and covered three domains: coding, problem-solving, and algorithmic thinking. The test was administered both before (pre-test) and after (post-test) the intervention. The control group ($n = 45$) received instruction through traditional methods, while the experimental group ($n = 45$) was trained using the gamified platform Blockland.kz.

To ensure clarity, the data are presented as group means with standard deviations, followed by calculated learning gains (Δ) for each skill domain.

Table 9. Comparison of pre-test and post-test scores between groups (*n* = 90)

Skill Domain	Control Group (*n* = 45)			Experimental Group (*n* = 45)		
	Pre-test (M ± SD)	Post-test (M ± SD)	Δ (Gain)	Pre-test (M ± SD)	Post-test (M ±SD)	Δ (Gain)
Coding	56.2 ± 2.8	63.5 ± 3.1	+7.3	56.4 ± 2.9	85.2 ± 3.4	+28.8
Problem-Solving	55.4 ± 2.9	64.2 ± 3.3	+8.8	57.0 ± 3.1	87.1 ± 2.9	+30.1
Algorithmic Thinking	57.0 ± 3.0	65.1 ± 2.8	+8.1	56.5 ± 2.7	86.4 ± 3.0	+29.9

As shown in Table 9, the control and experimental groups began the study with nearly identical pre-test scores, clustered between 55 and 57 points across all domains, suggesting equivalent baseline proficiency. Following the intervention, however, the experimental group demonstrated markedly larger improvements. Independent-samples t-tests on learning gains confirmed statistically significant differences between groups: Coding t(88) = 31.35, $p < 0.001$, $d = 6.61$, 95% CI [20.14, 22.86]; Problem-Solving $t(88) = 32.52$, $p < 0.001$, $d = 6.86$, 95% CI [20.00, 22.60]; Algorithmic Thinking $t(88) = 35.64$, $p < 0.001$, $d = 7.51$, 95% CI [20.58, 23.02]. These findings provide robust evidence that the experimental condition produced substantial learning benefits beyond the considerable gains observed in the control group. Participants who engaged with gamified programming content via the Blockland.kz platform consistently achieved post-test scores ranging from 85 to 87 points across all skill categories, far exceeding the results observed in the control group

Fig. 1 visualizes the mean pre-test and post-test scores for the control and experimental groups, averaged across coding, problem-solving, and algorithmic thinking. While both groups began at similar baselines, the experimental group showed markedly higher post-test performance. These visual patterns correspond with the unified inferential statistics in Table 10, which confirm the robustness and magnitude of the effects.

Mean pre-test and post-test scores (±SE) for control and experimental groups across coding, problem-solving, and algorithmic thinking combined. Fig. 1 shows that both groups began at comparable baselines (~56 points), but the experimental group achieved substantially higher post-test scores (~86 points), consistent with the statistical results reported in Table 10.

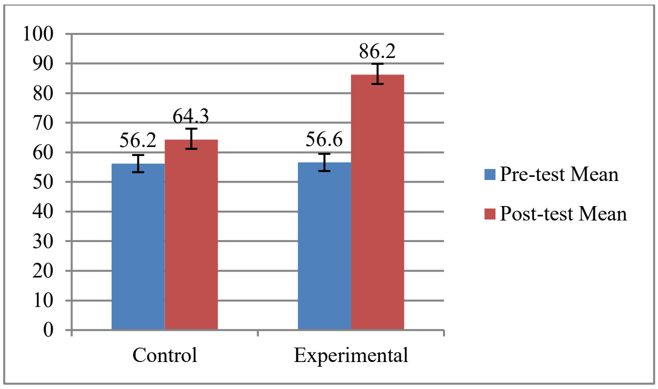

Fig. 1. Group score comparison (source: authors).

Independent-samples *t*-tests (df = 88) for all three skills—Coding Gain ($t(88) = 31.35$), Problem-Solving Gain ($t(88) = 32.52$), and Algorithmic Thinking Gain ($t(88) = 35.64$)—were extremely large in magnitude, exceeding conventional thresholds for statistical significance. This indicates clear and robust differences between the control and experimental groups in terms of skill gains. Specifically, the mean differences were 21.5 points for Coding, 21.3 points for Problem-Solving, and 21.8 points for Algorithmic Thinking. When paired with relatively low standard errors (0.686, 0.655, and 0.612, respectively), these results strongly reinforce the reliability of the observed effects.

Table 10 consolidates the results, reporting means, standard deviations, mean differences, t-values, p-values, effect sizes, and 95% confidence intervals in a single place. According to Table 10, the p-values for Coding Gain

($p < 0.001$), Problem-Solving Gain ($p < 0.001$), and Algorithmic Thinking Gain ($p < 0.001$) confirm that the improvements in the experimental group were not due to chance. Effect sizes were exceptionally large (Cohen's d = 6.61–7.51) with narrow confidence intervals, underscoring the robustness of the findings.

Table 10. Independent-samples t-test results for gain scores (df = 88)

Skill Domain	Control Group (n=45) (M ± SD)	Experimental Group (n=45) (M ± SD)	Mean Difference (Experimental–Control)	t (88)	p	Cohen's d	95% CI
Coding	7.3 ± 3.1	28.8 ± 3.4	21.5	31.35	< 0.001	6.61	[20.14, 22.86]
Problem-Solving	8.8 ± 3.3	30.1 ± 2.9	21.3	32.52	< 0.001	6.86	[20.00, 22.60]
Algorithmic Thinking	8.1 ± 2.8	29.9 ± 3.0	21.8	35.64	< 0.001	7.51	[20.58, 23.02]

Assumption checks conducted in SPSS v28 showed that the distribution of gain scores was approximately normal (Shapiro–Wilk test, Q–Q plots) and that homogeneity of variances was met (Levene's test, $p > 0.05$).

V. Discussion

To attain the primary aim of this study, i.e., to assess the effectiveness of gamification in a programming course (taking Python as an example) and its impact on the academic performance and proficiency of future computer science teachers the results suggest that gamification was associated with higher short-term learning gains and positive student perceptions. These findings are consistent with those of earlier research [22, 24], which also reported improvements in motivation and performance when gamification was implemented in programming learning.

The quasi-experimental results showed that the experimental group possessed considerably higher post-test scores in coding, problem-solving, and algorithmic thinking compared to the control group. In the meantime, survey data from a larger group of students indicated that students perceived gamification elements—such as points, badges, and leaderboards—to be strong motivators that assisted in fostering engagement and persistence in programming tasks. Approximately 45% of the participants explicitly stated that rewards and achievements for finishing assessments assisted in enhancing their desire to learn.

Taken together, these findings indicate an alignment between self-reported motivation and observed performance outcomes. The survey evidence on heightened motivation and engagement corroborates well with the experimental group's improved performance. Students who were more motivated by gamification were more likely to persevere on tricky problems, thereby developing more sophisticated problem-solving and algorithmic thinking skills. At the same time, the survey also reflected concerns of competition and overreliance on extrinsic motivators, reminding us that these positive effects are not uniform and have possible negative consequences if gamification design is not properly balanced.

By integrating both data sets, the study therefore presents a more nuanced view: gamification appears linked to measurable learning gains and is also associated with reported increases in engagement and confidence. But the ambivalent attitudes toward gamification mechanics suggest that future work should investigate how specific elements differentially affect motivation and performance in diverse learners.

VI. Conclusion

The study examined the potential of gamification for enhancing programming skills and academic achievements among future Informatics teachers. This study assessed gamification's impact on computer science teacher education courses. In a quasi-experimental condition (n = 90), students in the gamified condition made significantly greater gains in coding, problem-solving, and thinking algorithmically than traditionally instructed students. A concurrent survey of 203 students also confirmed that game elements such as rewards, badges, and leaderboards were perceived to be engaging and motivating.

Together, these results suggest that gamification can enhance academic performance and learner motivation in programming courses. However, the disparity in outcomes also indicates that traditional approaches may lack interactive and engaging elements necessary to sustain student interest and achievement.

A. Limitations and Future Research Directions

While the findings of this research support gamification as potentially effective in enhancing performance and skill acquisition in programming topics, there are certain limitations to be mentioned:

Sample Size and Generalizability: The study would have been limited by the sample size or by the specific condition under which it was conducted (e.g., prospective Informatics teachers from selected universities). Thus, the generalizability of the findings to other groups—e.g., students from other disciplines, institutions, or levels of education could be restricted. Age, prior experience with programming, and various motives for studying computer science can also influence learning outcomes and attitudes toward gamification. Moreover, the quasi-experimental design did not involve random assignment, and the evidence was drawn from two distinct samples (n = 90 in the experiment and n = 203 in the survey), which limits direct comparability of results. Although ANCOVA was used to adjust for the significant baseline difference in problem-solving pre-test scores, future research should employ random assignment to minimize such initial group imbalances.

Short-Term Assessment: The study concentrated primarily on the short-term learning gains, such as test scores and reported motivation. However, it did not measure long-term retention of skills, a central aspect of programming education. Future studies may conduct longitudinal analysis to determine whether the positive effects of gamification are stable over the long run.

Sustainability of outcomes: Another limitation is how long the intervention lasted. The study spanned a single 15-week semester, and the measures were collected immediately upon completion of the course. While the results show immediate short-term gains in coding, problem-solving skills, and algorithmic thinking capacity and motivation, the absence of long-term longitudinal follow-up data leaves us wondering if these gains persist in the long run. Furthermore, although the instrument aligned with course objectives, more comprehensive evidence of its validation and inter-rater reliability would strengthen confidence in the robustness of the measures.

Developer bias: Because the gamified platform Blockland.kz was developed by the authors, a potential source of bias must be acknowledged. At the same time, this involvement ensured a strong alignment of the platform with pedagogical objectives, though future studies should consider independent evaluations to further strengthen the evidence base.

Potential novelty effect: The enhancements in students' engagement and achievement that are found can be partly attributed to the novelty of gamification and less to the design of the gamified elements. Once novelty has worn off, the degree of engagement can dissipate. Additional research must investigate the long-term maintenance of engagement. Because gamification represented a new and engaging instructional approach for many participants, part of the observed improvement in skills and motivation may reflect students' short-term enthusiasm for the unfamiliar method rather than its inherent pedagogical value. This effect could inflate the outcomes, particularly since measurements were taken only at the end of the 15-week intervention, when the sense of novelty may still have been strong. Therefore, the use of mixed designs where one group receives gamification after already having prior exposure reduces the novelty factor.

Statistical assumptions: While parametric assumptions were satisfied in this study (normality and homogeneity of variances), replication using non-parametric tests or alternative statistical approaches across different samples would provide additional confirmation of the robustness of the findings.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Aigul Aldabergenova, Ulzhalgas Yesseikyzy, and Aiym Yesseikyzy were responsible for the study design, experiment implementation, and drafting of the initial manuscript. Assem Yerkinova contributed to the interpretation of the results and refinement of the text. Ilyas Yessengabylov carried out the scientific editing. All authors have read and approved the final version of the manuscript.

References

[1] 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," *Comput. Educ.: Artif. Intell.*, vol. 3, 100096, 2022. doi: 10.1016/j.caeai.2022.100096

[2] C. A. Bodnar, D. Anastasio, J. A. Enszer, and D. D. Burkey, "Engineers at play: Games as teaching tools for undergraduate engineering students," *J. Eng. Educ.*, vol. 105, no. 1, pp. 147–200, 2016. doi: 10.1002/jee.20109

[3] C. Y. Tsai, H. Lin, and S. C. Liu, "The effect of pedagogical GAME model on students' PISA scientific competencies," *J. Comput. Assist. Learn.*, vol. 36, no. 3, pp. 359–369, 2020. doi: 10.1111/jcal.12401

[4] A. Domínguez *et al.*, "Gamifying learning experiences: Practical implications and outcomes," *Comput. Educ.*, vol. 63, pp. 380–392, 2013. doi: 10.1016/j.compedu.2012.12.020

[5] P. Fotaris and T. Mastoras, "Escape rooms for learning: A systematic review," *Comput. Educ.*, vol. 146, 103682, 2019. doi: 10.34190/GBL.19.179

[6] Z. Zainuddin, S. K. W. Chu, M. Shujahat, and C. J. Perera, "The impact of gamification on learning and instruction: A systematic review of empirical evidence," *Educ. Res. Rev.*, vol. 30, 100326, 2020. doi: 10.1016/j.edurev.2020.100326

[7] N. A. Ermakan, B. S. Ualikhanova, B. K. Rakhashev, and S. A. Ramazanova, "Effectiveness of the gamification method in the process of phenomenon-based learning in secondary school (in the example of Kazakhstan)," *J. Ecohumanism*, vol. 3, no. 8, pp. 1330–1344, 2024. doi: 10.62754/joe.v3i8.4814

[8] D. S. Alsuhaymi and O. M. Alotaibi, "Gamification's efficacy in enhancing students' HTML programming skills and academic achievement motivation," *J. Educ. E-Learn. Res.*, vol. 10, no. 3, pp. 397–407, 2023. doi: 10.20448/journal.509.2023.103.397.407

[9] S. Schöbel *et al.*, "A research agenda for the why, what, and how of gamification designs: Outcomes of an ECIS 2019 panel," *Commun. Assoc. Inf. Syst.*, vol. 46, pp. 706–721, 2020. doi: 10.17705/1CAIS.04630

[10] K. D. Cuervo-Cely, "Effect of gamification on the motivation of computer programming students," *J. Inf. Technol. Educ., Res.*, vol. 21, pp. 1–23, 2022. doi: 10.28945/4917

[11] M. Videnovik, T. Vold, L. Kiønig, A. M. Bogdanova, and V. Trajkovik, "Game-based learning in computer science education: A scoping literature review," *Int. J. STEM Educ.*, vol. 10, no. 1, p. 54, 2023. doi: 10.1186/s40594-023-00447-2

[12] D. R. Joshi, K. P. Adhikari, B. Khanal, J. Khadka, and S. Belbase, "Behavioral, cognitive, emotional and social engagement in mathematics learning during COVID-19 pandemic," *PLoS ONE*, vol. 17, no. 11, 2022. doi: 10.1371/journal.pone.0278052

[13] G. J. Hwang, L. Y. Chiu, and C. H. Chen, "A contextual game-based learning approach to improving students' inquiry-based learning performance in social studies courses," *Comput. Educ.*, vol. 81, pp. 13–25, 2015. doi: 10.1016/j.compedu.2014.09.006

[14] B. Kaldarova *et al.*, "Applying game-based learning to a primary school class in computer science terminology learning," *Front. Educ.*, vol. 8, 2023. doi: 10.3389/feduc.2023.1100275

[15] M. Tsur and N. Rusk, "Scratch microworlds: Designing project-based introductions to coding," in *Proc. 49th ACM Tech. Symp. Comput. Sci. Educ.*, 2023, pp. 894–899. doi: 10.1145/3545945.3569764

[16] S. Subhash and E. A. Cudney, "Gamified learning in higher education: A systematic review of the literature," *Comput. Hum. Behav.*, vol. 87, pp. 192–206, 2018. doi: 10.1016/j.chb.2018.05.028

[17] F. Azzali, A. Abas, and R. Din, "Enhancing usability in gamified computer programming courses for higher education," *Prog. Comput. Learn.*, vol. 2, no. 1, p. 11, 2025. doi: 10.37934/picl.2.1.111

[18] A. M. Barua and S. S. Bharali, "Gamification and its challenges in e-learning: A case study of computer science learners in KKHSOU," *Asian Assoc. Open Univ. J.*, vol. 18, no. 3, pp. 233–245, 2023. doi: 10.1108/AAOUJ-06-2023-0066

[19] R. N. Landers, "Developing a theory of gamified learning: Linking serious games and gamification of learning," *Simul. Gaming*, vol. 45, no. 6, pp. 752–768, 2014. doi: 10.1177/1046878114563660

[20] R. Smiderle, S. J. Rigo, L. B. Marques, J. A. P. M. Coelho, and P. A. Jaques, "The impact of gamification on students' learning, engagement and behavior based on their personality traits," *Smart Learn. Environ.*, vol. 7, no. 1, p. 3, 2020. doi: 10.1186/s40561-019-0098-x

[21] D. Codish and G. Ravid, "Personality based gamification-Educational gamification for extroverts and introverts," in *Proc. 9th CHAIS Conf. Study Innov. Learn. Technol.*, vol. 1, pp. 36–44, 2014.

[22] S. Kiraly and T. Balla, "The effectiveness of a fully gamified programming course after combining with serious games," *Acta Didactica Napocensia*, vol. 13, no. 1, pp. 65–76, 2020. doi: 10.24193/adn.13.1.7

[23] J. Swacha and J. Szydłowska, "Does gamification make a difference in programming education? Evaluating FGPE-supported learning outcomes," *Educ. Sci.*, vol. 13, no. 10, 2023. doi: 10.3390/educsci13100889

[24] B. Marín, J. Frez, J. Cruz-Lemus, and M. Genero, "An empirical investigation on the benefits of gamification in programming courses," *ACM Trans. Comput. Educ.*, vol. 19, no. 1, pp. 4:1–4:22, 2018. doi: 10.1145/3231711

[25] F. Pradana, P. Setyosari, S. Ulfa, T. Hirashima, and F. Amalia, "The effect of gamification-based programming e-learning platform on students' achievement," in *Proc. 8th Int. Conf. Sustain. Inf. Eng. Technol.*, 2023, pp. 370–376. doi: 10.1145/3628232.3628285

[26] T. B. Durmaz, J. L. Fuertes, and R. Imbert, “Influence of gamification elements on explicit motive dispositions,” *IEEE Access*, vol. 10, pp. 118058–118071, 2022. doi: 10.1109/ACCESS.2022.3224696

[27] E. Adam and R. van den Berg, “The influence of gamified e-learning quizzes on students’ motivation–a case of programming students at a South African higher education institution,” *Indep. J. Teach. Learn.*, vol. 17, no. 2, 2022. doi: 10.17159/nfzqpd19

[28] J. N. Proulx, M. Romero, and S. Arnab, “Learning mechanics and game mechanics under the perspective of self-determination theory to foster motivation in digital game based learning,” *Simul. Gaming*, vol. 48, no. 1, pp. 81–97, 2017. doi: 10.1177/1046878116646915

[29] V. Jusas, D. Barisas, and M. Jančiukas, “Game elements towards more sustainable learning in object-oriented programming course,” *Sustainability*, vol. 14, no. 4, 2022. doi: 10.3390/su14042325

[30] D. Vlachopoulos and A. Makri, “The effect of games and simulations on higher education: A systematic literature review,” *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, p. 22, 2017. doi: 10.1186/s41239-017-0062-1

[31] O. S. Kaya and E. Ercag, “The impact of applying challenge-based gamification program on students’ learning outcomes: Academic achievement, motivation and flow,” *Educ. Inf. Technol.*, vol. 28, no. 8, pp. 10053–10078, 2023. doi: 10.1007/s10639-023-11585-z

[32] L. Sun and J. Liu, “Effects of gamified Python programming on primary school students’ computational thinking skills: A differential analysis of gender,” *J. Educ. Comput. Res.*, vol. 62, no. 3, pp. 846–874, 2024. doi: 10.1177/07356331231217056

[33] Y. C. Chen, C. Y. Chao, and H. T. Hou, “Learning pattern recognition skills from games: Design of an online pattern recognition educational mobile game integrating algebraic reasoning scaffolding,” *J. Educ. Comput. Res.*, vol. 61, no. 6, pp. 1232–1251, 2023. doi: 10.1177/07356331231188582

[34] J. Hamari, D. J. Shernoff, E. Rowe *et al*., “Challenging games help students learn: An empirical study on engagement, flow, and immersion in game-based learning,” *Comput. Hum. Behav.*, vol. 54, pp. 170–179, 2016. doi: 10.1016/j.chb.2015.07.045

[35] K. Czakóová, “Developing algorithmic thinking by educational computer games,” in *Proc. eLearning Softw. Educ. (eLSE)*, vol. 1, pp. 26–33, 2020.