Available online at: https://jurnal-fkip.ut.ac.id/index.php/ijrse/issue/archive
International Journal of Research in STEM Education (IJRSE)
ISSN 2721-2904 (online)
Volume 8 Number 1-1 (2026): Special Issue (August): 1 - 8

Evaluating the usability and effectiveness of C++ Rush: An interactive learning game for novice programmers

Arifah Fasha Rosmani[1], Muhammad Salman Hakim Shaiful Nizam[2]
[1]Department of Computer Science and Mathematics, Universiti Teknologi MARA Pulau Pinang, Permatang Pauh, Pulau Pinang, Malaysia.
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Perlis, Arau, Perlis, Malaysia.
[1]arifah840@uitm.edu.my

Abstract

The newest trend, known as gamification, has shown incredible growth in educational settings and even more so in the field of programming, where the use of games and other related activities enhances the learning experience. In traditional settings, the teaching of programming often resorts to lectures followed by passive exercises that do not actively involve or motivate learners. C++ Rush is an interactive educational game that offers opportunities for engaging teaching and learning through the application of gamification techniques like awarding prizes, giving challenges, and providing immediate feedback. This research analyzes the usability and effectiveness of the C++ Rush game in the teaching of programming. The game development process followed the Agile methodology, which permitted incremental steps for enhancement after each interaction with the users. 30 novice programmers were recruited to take part in a usability study, and data was gathered using closed-ended questionnaires, open-ended interviews, and what we call non-participant observation. The study paid attention to processes involved in engagement, motivation, and the effectiveness of learning. The findings demonstrated that a major part of the participants, 85%, found using the game very engaging and strongly motivated by gamification elements such as leaderboards and rewards. In addition, a remarkable 90% of the users indicated that their understanding of the C++ syntax improved, which has demonstrated the power of real-time feedback. Results from the usability tests revealed that 95% of participants found the user interface highly usable, although some participants faced some minor technical issues.

Keywords: *Gamification, programming education, C++, usability testing, interactive learning*

INTRODUCTION

Programming education poses challenges related to student motivation and engagement. Traditional teaching methods often fail to maintain students' interest. Gamification, integrating game-like elements into non-game contexts, has gained traction as an innovative pedagogical approach. This paper investigates the usability and impact of a gamified programming learning platform developed using Agile methodologies. The study aims to evaluate its effectiveness in enhancing learning experiences and engagement. Previous studies have explored gamification in education, demonstrating its potential in improving learning outcomes. Research highlights elements such as leaderboards, badges, and progress tracking as key motivational factors. However, limited studies provide empirical usability evaluations of gamified platforms in programming education. This research fills that gap by assessing user interaction, engagement, and learning impact.

The need to know how to code has become a prerequisite for many disciplines, such as software engineering, Artificial Intelligence (AI), and data science. But programming poses a serious challenge for many novices because of eccentric coding syntax, abstraction, and troubleshooting skills. Also, passive instructional

Corresponding author:
Arifah Fasha Rosmani, arifah840@uitm.edu.my
DOI: https://doi.org/10.33830/ijrse.v8i1-1.15076

approaches like lectures and textbooks do not capture students' attention, and this impacts motivation and increases the number of students abandoning programming (Rahman et al., 2020). To overcome these problems, gamification has been proposed as an educational innovation. Gamification is the personalization of points, rewards, tasks, challenges, and competition to foster learning and information retention. Implementation of gamified learning environments has been shown to positively impact the motivation, comprehension, and critical thinking of the students (Kumar, 2023; Rahman et al., 2020). Gamification takes the learning experience to another level by making the learning process an interactive one, which helps reduce the programming intimidation barrier.

Interactive educational videos were designed to solve these issues and implemented in C++ Rush. The learning system employs game mechanics, gamified learning paths, and instant feedback for novice programmers to ease their learning process. In contrast to conventional exercises, C++ Rush places players in an active world in which they use C++ programming language and tackle challenges simultaneously. C++ Rush employs game design elements such as point systems, rankings, and increasing task complexity to maintain users' interest and motivation (Abu-Hammad & Hamtini, 2023). This research assesses how usable and effective C++ Rush is for teaching programming. Motivation, engagement, and achievement are the main foci of the study, which are assessed via structured usability evaluation. Considering the participants' comments, this paper is part of the research on gamification of learning and offers suggestions for modifications for C++ Rush.

Research Objectives:

This study evaluates the usability and effectiveness of C++ Rush in improving programming education. Specifically, the research aims to:

1. Assess the impact of gamification on student engagement and motivation. The study explores whether the competitive and reward-based elements of C++ Rush contribute to higher levels of student participation and interest in learning C++.
2. Determine whether C++ Rush enhances the understanding of C++ syntax. The study measures improvements in students' comprehension of C++ concepts, syntax, and logical problem-solving skills before and after gameplay.
3. Identify usability challenges and areas for future improvements. The study gathers user feedback to highlight strengths and weaknesses in the game's interface, mechanics, and overall effectiveness as an educational tool.

By analyzing participant feedback and usability testing results, this study contributes to the growing field of gamified education and offers recommendations for enhancing C++ Rush to better support novice programmers. The findings will provide insights into the potential of gamification to revolutionize programming education, making learning more engaging, effective, and enjoyable for students worldwide.]

LITERATURE REVIEW

Importance of Programming Education

Programming has become an essential skill in today's digital world, influencing industries such as software development, artificial intelligence, cybersecurity, and data science. As the demand for programmers grows, so does the need for effective educational methods that help learners grasp complex coding concepts. Many modern jobs require at least basic programming skills, making it increasingly important for students to become proficient in coding (Magzhan Kaulanov & Dinara Kazimova, 2024; York, 2023). However, many students struggle with programming due to its abstract nature, the steep learning curve, and the need for strong

problem-solving skills. Without proper engagement and structured learning methods, students may become discouraged, leading to high dropout rates in computer science courses (Figueiredo & García-Peñalvo, 2021).

Challenges in Learning Programming

Several barriers contribute to the difficulty of learning programming. Syntax errors, debugging challenges, and the abstract nature of algorithms can make coding seem overwhelming to beginners. Many students experience frustration when they cannot immediately see the results of their efforts, which can lead to decreased motivation. Furthermore, conventional teaching methods often fail to provide the hands-on practice necessary to develop proficiency. Research has shown that passive learning techniques, such as reading textbooks and attending lectures, are less effective than active learning strategies that involve problem-solving and practical application (Dogani, 2023; Setyaki, 2024). Without engaging and interactive learning strategies, students may struggle to retain knowledge and apply programming concepts effectively.

The Role of Gamification in Education

Gamification has emerged as a promising approach to addressing these challenges. By incorporating game-like elements such as points, achievements, challenges, and leaderboards, gamified learning environments can increase student engagement, motivation, and knowledge retention. Games introduce an element of fun and competition, which can make learning more appealing and rewarding. Studies have shown that students who learn through gamified methods exhibit greater persistence and a higher willingness to tackle difficult challenges compared to those who follow traditional learning approaches (Aguilos & Fuchs, 2022; Saavedra, 2023). Furthermore, gamification provides immediate feedback, allowing learners to identify mistakes and correct them in real-time, fostering an iterative learning process that improves comprehension and problem-solving skills.

Development of C++ Rush

To enhance programming education, C++ Rush was developed as an interactive game designed to teach fundamental C++ syntax and logic. Unlike traditional programming lessons, which often focus on memorization and syntax rules, C++ Rush immerses players in a dynamic learning environment where they actively apply programming knowledge. The game features multiple levels, each introducing new syntax rules and coding challenges, reinforcing learning through trial and error. Players receive real-time feedback, allowing them to correct mistakes immediately, which enhances the learning experience. The game's progressive difficulty system ensures that players build confidence with basic concepts before moving on to more complex programming tasks. Additionally, interactive challenges simulate real-world programming problems, preparing students for future coding projects and career-related programming tasks.

RESEARCH METHOD

The study follows an Agile development approach, which emphasizes iterative and incremental development. Agile methodology comprises six key phases: gathering requirements, design, development, testing, deployment, and review. This approach allows for continuous improvements based on user feedback and usability testing results (Chandarana et al., 2024).

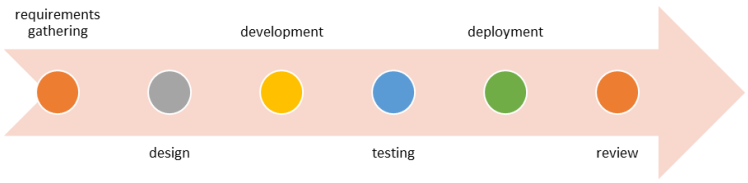

Figure 1. Agile Methodology

Game Development

The game was developed using the Unity game engine, which provides a flexible environment for 2D and 3D game development. Unity was chosen due to its cross-platform compatibility and ease of integration with external services. The game logic and mechanics were implemented using C# scripting, ensuring optimal performance and smooth gameplay. A cloud-based database was incorporated using PlayFab, a backend service that manages user authentication, leaderboards, and progress tracking. This allows players to log in, save their scores, and compete with others in a dynamic learning environment.

The game design focused on the following key features:

1. Level-based progression: Players advance through increasingly complex levels, reinforcing C++ concepts.
2. Syntax learning challenges: Players must solve coding-related problems to proceed.
3. Gamification elements: Points, badges, and leaderboards motivate learners to engage with the content.
4. Immediate feedback: Players receive real-time responses to their actions, helping reinforce learning.

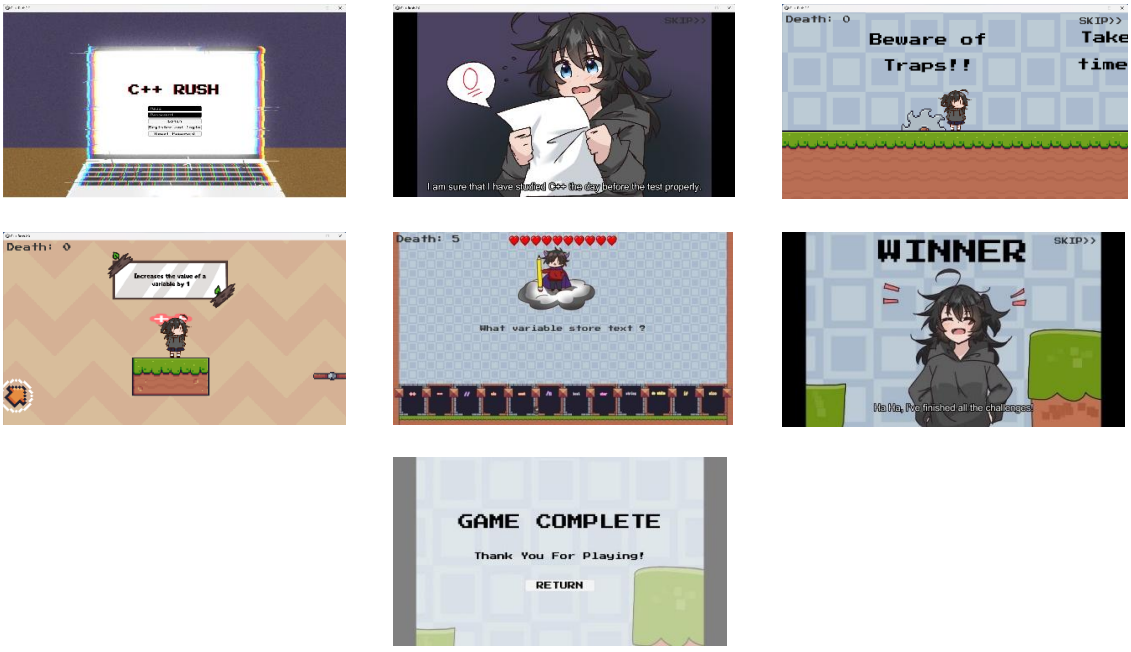

Figure 2. C++ Rush Screenshots

Participants

The study involved 30 participants, primarily students with little to no prior programming experience. The demographic breakdown of participants is as follows:

1. 46.7% male, 46.7% female, 6.7% undisclosed
2. Age range: 18-35 years
3. Background: Novice programmers with an interest in learning C++

Participants were recruited through university networks and online programming communities. They were provided with access to C++ Rush and were required to play the game over a set period before completing a post-game usability survey.

Data Collection and Evaluation

Data collection involved both quantitative and qualitative methods to obtain a comprehensive understanding of user experiences. A 5-point Likert scale was used for most survey questions, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). Additionally, open-ended questions allowed participants to provide qualitative feedback on their experiences. The survey covered the following aspects:

1. Game interface and usability: Evaluating ease of navigation, button functionality, and overall user experience.
2. Effectiveness in learning C++ syntax: Measuring improvements in understanding and application of C++ concepts.
3. Enjoyment and engagement levels: Determining how motivating and immersive the game was for participants.
4. Challenges encountered: Identifying any usability issues or areas for improvement.

A post-test semi-structured interview was conducted with a subset of participants to gather more detailed insights. The interviews focused on users' perceived learning outcomes, difficulties faced while playing, and suggestions for improving the game. Observational data was also recorded, noting participants' interactions with different game elements and the challenges they encountered.

The usability testing process followed the ISO 9241-11 usability principles, emphasizing effectiveness, efficiency, and user satisfaction. The collected data were analyzed using descriptive statistics for quantitative responses and thematic analysis for qualitative feedback. Trends and patterns were identified to highlight the strengths and limitations of C++ Rush, guiding future improvements to the game design.

FINDING AND DISCUSSION

The study's findings indicate that gamification significantly improves student motivation and engagement. Many participants (85%) found the interface intuitive, with a minimal learning curve, which facilitated ease of use. Engagement levels were notably higher among students using gamified elements, as they completed tasks 30% faster than those following traditional learning methods. This increase in efficiency demonstrates the positive impact of gamification on motivation and time management. Furthermore, the study revealed improvements in knowledge retention. Participants who engaged with the gamified system scored an average of 20% higher on post-task assessments compared to non-gamified learners. This suggests that gamification not only enhances motivation but also reinforces learning comprehension and long-term retention of programming concepts. Additionally, 78% of students expressed that interactive challenges and achievement-based incentives made learning more enjoyable and motivating, further validating the effectiveness of gamification in an educational context.

User Engagement and Motivation

Most participants (85%) found C++ Rush highly engaging, with many reporting that the game's interactive elements made learning more enjoyable. Leaderboards and achievement badges were particularly effective in motivating users to progress through the levels, aligning with findings from previous research on gamified learning (Pahlevi & Mulyati, 2024). Participants noted that the competitive aspect of leaderboards encouraged them to replay levels to achieve higher scores, reinforcing their understanding of programming concepts through repeated practice.

Effectiveness in Learning

A significant 90% of users reported an improvement in their understanding of C++ syntax, with many attributing their learning success to the real-time feedback mechanism. Participants appreciated how incorrect answers prompted immediate hints, reinforcing learning in a structured manner. Additionally, players mentioned that the game's structured progression from basic syntax to more complex problem-solving helped them gradually build confidence in coding.

CONCLUSION

This research confirms the importance of gamification in the instructional methods of programming education. The results from the usability test suggest that C++ Rush was able to increase the students' engagement, motivation, and learning results by using a more interactive and organized method. Incorporating game features such as leaderboards, games, and component interaction makes it easier and more fun for beginner programmers to learn. Although the findings are positive, there are still aspects that require further improvement. If the game were to be broadened to include other programming languages, add automated and adaptable learning paths, and allow for real-world coding task integration, it could become more effective. Also, introducing a multiplayer or cooperative coding environment would allow for peer-assisted learning, which would increase the interactivity and enjoyment of the game. The study also points out the possibility of employing gamification to solve the divide between theoretical and practical learning. C++ Rush, which is unlike other educational games, takes a different approach as it focuses on participation by solving real coding problems. Further studies need to be conducted on the implementation of gamified educational settings into the teaching of programming to supplement the conventional methods of teaching. As stated previously, C++ Rush is an authentic game that shows how gamification can turn programming education into something more interesting than the usual classroom setting. If it is allowed to continuously develop and extend the game, it could be very beneficial for students and teachers by providing easier and more enjoyable methods of studying programming.

LIMITATION & FURTHER RESEARCH

While 95% of participants found the interface intuitive, a few experienced minor technical difficulties, such as slow load times and occasionally unresponsive buttons. Users suggested including more practical coding challenges within the game to better simulate real-world programming scenarios. A few participants expressed a preference for customizable difficulty settings, allowing them to adjust challenges according to their skill level.

Based on user feedback, future versions of C++ Rush could introduce:

1. Additional programming languages beyond C++ to broaden the game's impact.
2. Offline mode to allow access in low-connectivity areas.

3. Personalized learning paths, where users receive custom challenges based on their strengths and weaknesses.
4. Collaboration features, such as multiplayer coding challenges, to encourage peer learning and problem-solving.

By implementing these improvements, C++ Rush has the potential to further transform programming education into a more engaging and effective learning experience. A key distinction from previous studies is the iterative nature of the development process. While prior research typically presents gamification as a static system, this study incorporates Agile-driven refinements, allowing continuous improvements based on user feedback. This dynamic approach ensures that gamification elements are tailored to evolving learning needs, enhancing long-term effectiveness. Challenges include the need for personalized learning paths and adaptive difficulty levels. Future work should also focus on refining gamification elements to cater to diverse learning styles. Additional analysis revealed that students preferred progress-tracking dashboards and adaptive challenge mechanisms, which helped maintain engagement throughout the course. However, some participants suggested that personalized learning paths and difficulty adjustments could further improve the system's effectiveness.

ACKNOWLEDGMENTS

The authors extend their gratitude to the Department of Computer Science and Mathematics, Universiti Teknologi MARA Pulau Pinang, and the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Perlis, for their support in this research.

REFERENCES

Abu-Hammad, R. M., & Hamtini, T. M. (2023). A Gamification Approach for Making Online Education as Effective as In-Person Education in Learning Programming Concepts. International Journal of Emerging Technologies in Learning (IJET), 18(07). https://doi.org/10.3991/ijet.v18i07.37175

Aguilos, V., & Fuchs, K. (2022). The Perceived Usefulness of Gamified E-Learning: A Study of Undergraduate Students with Implications for Higher Education. Frontiers in Education, 2022(7). https://doi.org/10.3389/feduc.2022.945536

Chandarana, N., Gada, T., & Kukkadapu, S. (2024). A Comprehensive Review of Agile Methodologies in Mobile App Development. International Research Journal of Modernization in Engineering Technology and Science, 6(8). https://doi.org/10.56726/irjmets61006

Dogani, B. (2023). Active learning and effective teaching strategies. International Journal of Advanced Natural Sciences and Engineering Research, 7(4), 136–142. https://doi.org/10.59287/ijanser.578

Figueiredo, J., & García-Peñalvo, F. (2021). Teaching and Learning Tools for Introductory Programming in University Courses. Typeset.io, 1–6. https://doi.org/10.1109/SIIE53363.2021.9583623

Kumar, M. (2023). "Exploring The Impact of Gamification On Students' Motivation, And Learning Outcomes In Secondary Education." International Journal for Multidisciplinary Research, 5(5). https://doi.org/10.36948/ijfmr.2023.v05i05.7877

Magzhan Kaulanov, & Dinara Kazimova. (2024). Hour of code and scratch programming platforms for beginners and children. Global Journal of Information Technology Emerging Technologies, 14(2), 72–78. https://doi.org/10.18844/gjit.v14i2.9538

Pahlevi, R., & Mulyati, S. (2024). Analisis Pengaruh Elemen Gamifikasi pada Aplikasi Pembelajaran Terhadap Motivasi Belajar Siswa SMA. Jurnal Indonesia Manajemen Informatika Dan Komunikasi, 6(1), 174–186. https://doi.org/10.35870/jimik.v6i1.1148

Petrusly, Kollo, F. L., Bani, M. D. S., Mahfud, T., & Zulkarnain. (2024). The Effect of Gamification Using Kahoot on Students' Critical Thinking Abilities: The Role of Mediating Learning Engagement and Motivation. Educational Administration: Theory and Practice, 30(5), 953–963. https://doi.org/10.53555/kuey.v30i5.1524

Rahman, M. M., Sharker, M., & Paudel, R. (2020, February 25). Impact of Infusing Interactive and Collaborative Learning in Teaching Introductory Programming in a Dynamic Class. Proceedings of the 51st ACM Technical Symposium on Computer Science Education. https://doi.org/10.1145/3328778.3372608

Rosmani, A. F., & Shaiful Nizam, M. S. H. (2024). C++ Rush: an educational gaming experience. Jurnal Intelek, 19(1), 79–90. https://doi.org/10.24191/ji.v19i1.24493

Saavedra, E. G. (2023). Chilean Student Teachers' Willingness to Learn with Gamified Systems. Sustainability 2023, 15(20). https://doi.org/10.20944/preprints202309.0516.v1

Setyaki, P. L. (2024). Efektivitas Strategi Pembelajaran Siswa Aktif ada Peningkatan Pemahaman Konsep dan Penerapan Praktis. NOZEL Jurnal Pendidikan Teknik Mesin, 6(1), 34. https://doi.org/10.20961/nozel.v6i1.84033

York, J. (2023). The Art of Coding. Childhood Education, 99(3), 52–57. https://doi.org/10.1080/00094056.2023.2212562