Universiti Teknologi MARA
Cawangan Perlis
Jurnal Intelek

Volume 19 Issue 1 (February) 2024

C++ Rush: An Educational Gaming Experience

Arifah Fasha Rosmani[1*], Muhammad Salman Hakim Shaiful Nizam[2]
[1]Department of Computer Science and Mathematics, Universiti Teknologi MARA, Cawangan Pulau Pinang, Kampus Permatang Pauh, 13500 Permatang Pauh, Pulau Pinang, Malaysia.
[2]College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA, Cawangan Perlis, Kampus Arau, 02600 Arau, Perlis, Malaysia.

*Author's Email Address: *[1]arifah840@uitm.edu.my, [2]salmanhakim300@gmail.com*

Received Date: 30 October 2023
Accepted Date: 22 November 2023
Revised Date: 29 November 2023
Published Date: 31 January 2024
*Corresponding Author

ABSTRACT

Programming language is a subject that is quite difficult to understand at first glance, especially for beginners. Most beginners struggle to comprehend programming concepts because of their low maturity level, lack of programming experience, and difficulty understanding and learning the logic and scripting language. C++ is one of the basic, convenient programming languages that help programmers comprehend the principles of all programming languages. If the principles of C++ have been grasped, someone with a thorough knowledge of the language could easily transit to other programming languages. Therefore, this study proposes the combination of an educational game with C++ to provide early exposure to novice programmers and assist those who are struggling to learn the programming language. Furthermore, it attempts to create and develop an interactive, instructional programming game. Another purpose of this study is to test the usefulness of the generated programming game using a questionnaire. The study utilized the Agile Model because iterative development is one of its components. The agile technique consists of six (6) phases: requirements, design, development, testing, deployment, and review. This study did usability testing with 30 individuals. Most of them agreed that gamification might change a tough and complex topic into one that would be incredibly engaging. In addition, it has ultimately allowed users to properly learn C++. Future work may employ a broader range of programming languages.

Keywords: *C++, gamification, interactive games, novice learner, programming language*

INTRODUCTION

What exactly is a game? A game is an activity performed for enjoyment, sport, or competition. As indicated by Wan et al. (2017), modern electronic information technology and the internet have made computer games a source of entertainment. Nonetheless, they have adverse effects on college students, for instance, addiction. According to Seaton et al. (2018), educational games attempt to balance learning and enjoyment. Therefore, this study blends education and gaming into a single entity that will balance gaming and education through educational games. People must be encouraged to play educational games frequently because it is beneficial where the games help to facilitate people's comprehension of difficult topics, such as computer languages. This study will also assist individuals in comprehending the principles of the C++ programming language.

ISSN: 1675-9885 / E-ISSN 2231-7716
DOI: http://10.24191/ji.v19i1.24493

Educational games play an essential role in making life easier for learning. This is because the learning environment becomes engaging and appealing because more students participate in the educational challenge (Borna & Rad, 2018). Programming language is a subject that is quite difficult to understand at first glance, especially for novices. Most beginners struggle to comprehend Java's Oriented Object Programming (OOP) concept (Maiga, 2019), one of the programming languages that could be learned. As Zamin et al. (2018) stated, because of their low maturity level and lack of programming experience, novice programmers have difficulty understanding and learning the logic and scripting language.

Therefore, the objectives of this study are to design and develop an interactive educational game for programming and to evaluate the usability of the developed programming game. The proposed solution to the problem is to develop an interactive educational game for programming using the Unity game engine. This game focuses on learning the basic syntax of the C++ programming language and designing the interface of the game using Unity. The target users were novice programmers. This game included some information about C++ syntax, quizzes, and competitive environments between different users. It is hoped this study holds considerable significance and it could contribute to society in assisting novice programmers in understanding C++ by providing a better environment for learning programming languages using educational games. This would help the novice programmers to learn programming languages in a more fun way through educational games and assist them by giving them something to enjoy while learning the program.

This study has combined an educational game with C++, the most fundamental programming language, to provide early exposure to novice programmers and to assist those who are struggling to learn the programming language. Gamification uses game components, primarily video game aspects, in non-game contexts to increase motivation and engagement in learning. The use of gamification in a method or practice of teaching can help many learners who are frustrated by traditional teaching approaches (Alsawaier, 2018). In Malaysia, educational games regarding programming languages are relatively few. Some beginners who do not have a background in a programming language may find it difficult to grasp the concept at the start of the learning process (Baharum et al., 2020). To develop an interactive game, it is hoped that this game will make learning programming much more fun.

RELATED WORKS

C++ Computer Language

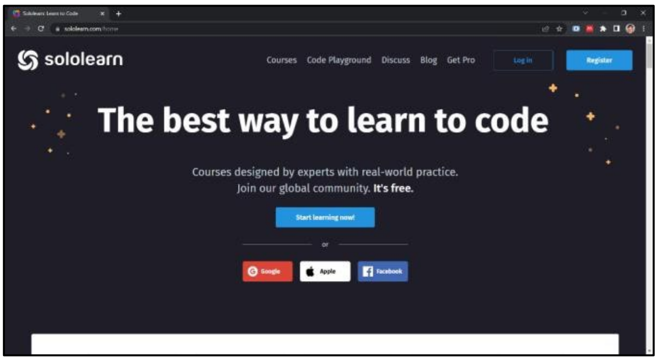
A computer language is any set of rules that transforms strings or graphical program elements (in the case of visual programming languages) into different types of machine code output. The two most popular high-level general-purpose programming languages, C and C++, each rely on a different concept and approach to programming design. For a very long time, these two languages have played and continued to play crucial roles in software creation (Yu He, 2009). C++ is a powerful and versatile programming language that encompasses multiple programming paradigms, making it a multi-paradigm language. It supports procedural, functional, object-oriented, and generic programming styles, providing developers with flexibility in designing and implementing their software solutions. Bjarne Stroustrup designed C++, and it first appeared in 1983, with development taking place at Bell Labs (Shaleynikov, 2017). In terms of typing discipline, C++ is characterized by static, nominative typing, with some level of type inference. This means that variable types need to be explicitly declared, enhancing code clarity, and catching potential errors during compilation. However, the language also allows for some type of inference, which can reduce the need for explicit type declarations in certain cases. C++ is platform-independent, making it compatible with various operating systems, including Linux, MacOS, and Solaris. This cross-platform capability contributes to its widespread adoption and usage in a variety of computing environments. When it comes to file extensions, C++ source code files can be identified by several extensions, including .cc, .cpp, .C, c++, .h, .hh, .hpp, .hxx, and .h++. These extensions are used for source code files and header files, each serving a specific purpose in the organization of C++ projects. In summary, C++ is a versatile programming language with a rich set of features and paradigms, allowing developers to create efficient and scalable software across different platforms. Its multi-paradigm nature, combined with static typing and cross-platform compatibility, makes it a popular choice for a wide range of applications.

Gamification

Gamification adapts game design components and the fundamental ideas and concepts that underpin gameplay to new situations. Gamification is popular at educational level where it attracts researchers and students from various fields such as education, information studies, human-computer interaction, and health. Despite extensive debate about its benefits and drawbacks, little empirical research has been conducted to establish gamification as a credible concept and to demonstrate its effectiveness as a tool for motivating and engaging users in non-entertainment settings (Seaborn & Fels, 2015). Related to this study, gamification is the process of incorporating game design elements into non-game contexts to make activities feel more like games (Sailer et al., 2017).

Gamification elements have two sides where it can both motivate learners and demotivate them. Reyssier et al. (2022) stated that game elements which were randomly chosen would typically demotivate learners. Although diverse effects were experienced by learners, a more thorough examination showed that gamification favored the least motivated students to perform arithmetic. Their starting level of motivation and their player type substantially impacted the fluctuation in motivation during doing the research. These effects differed depending on the game element they used. According to these findings, gamification should be adjusted to players' initial motivation levels for the learning task and player profiles to boost efficiency. Gamification elements used in this game were score and ranking.

Previous Works



The best way to learn to code

Courses designed by experts with real-world practice.
Join our global community. **It's free.**

Figure 1: Sololearn Homepage

Figure 1 shows a screenshot from the homepage of the Sololearn website (SoloLearn - Learn to Code, 2019). This website provides knowledge and information on all the programming languages that programmers need to know. Besides, the programmers can share their code here, and everyone can play around with it worldwide. Furthermore, this website has a mobile application, which makes it easy for people to use it wherever and whenever they like. Based on my observation, this website is not interesting to learn coding while having fun. The user needs to read the information beforehand and choose an answer from a multiple-choice question or fill in the blank with the correct coding.

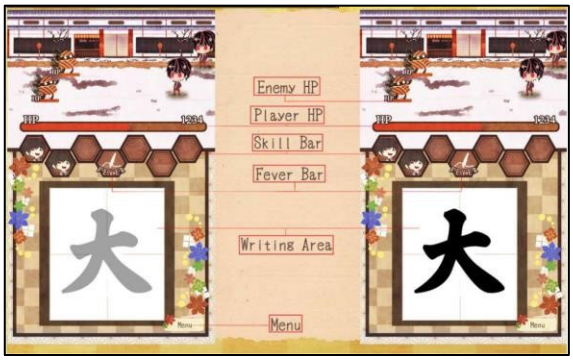


Figure 2: Gamification of Learning Kanji with "Musou Roman" Game

Anam Fathoni and Delima (2016) created their own learning game to make learning kanji much easier as depicted in Figure 2. The game uses a role-playing game that allows players to get into the interactive fiction story, enabling them to get quests, uncover secrets, communicate with the animated characters in the game, and earn rewards for finishing assigned tasks. Learning kanji has always been a difficult thing to do when learning Japanese because there are more than 2000 characters to memorize. This game helps to familiarize the players with all the available characters one at a time, thus developing their skills to learn effectively. Based on my observation, this is a good sample of gamification because it is easy and fun to learn.

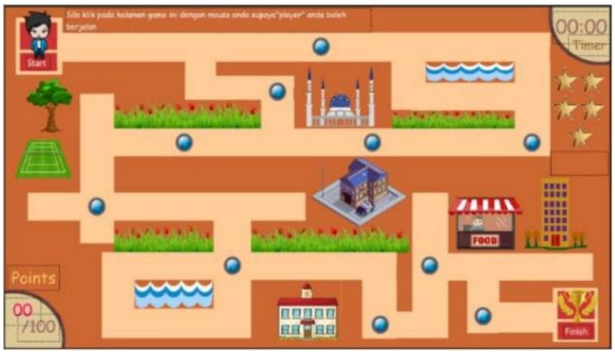

Sila klik pada halaman game ini dengan mouse anda supaya"player" anda boleh berjalan

Figure 3: Asmaul Husna Learning through Gamification and Adaptation of Signaling Principle

As illustrated in Figure 3, Rosmani and Zakaria (2018) also developed their gamification by providing information to Asmaul Husna. In this game, the goal is to reach the finishing line while collecting as much of the blue circle as possible in the maze, which then will give the user a question regarding Asmaul Husna; only the correct answer will be counted. Based on my observation, the game is fun, but it lacks the challenges of making the game more difficult.

Figure 4: Dragon box elements gameplay

Dragon Box Elements is a game in which players must gather an army to slay the wicked dragon Osgard and defend Euclid's island. This game as shown in Figure 4 is designed for players to learn everything they need to know about Euclid's theorems and fundamental geometry. A suitable example of a serious game is Dragon Box Elements, for those who enjoy playing video games while learning arithmetic. Based on my observation, this game is educational fun, and challenging.

METHODOLOGY

This study utilized iterative development, which is a component of the Agile development methodology. The methodology has been mapped based on the Agile model (Grepon, Benzar Glen, et al., 2021). As shown in Figure 5, this conceptual design includes requirement analysis, system design, algorithm design, the development phase, testing, and review. However, this paper is going to discuss only the development phase of the research.

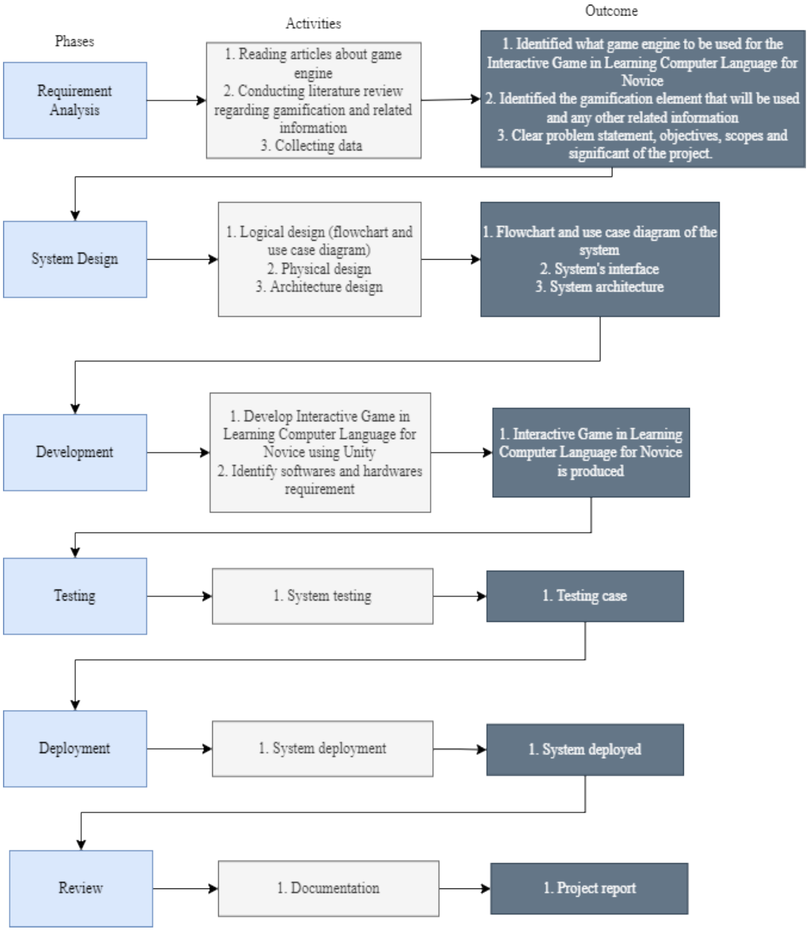
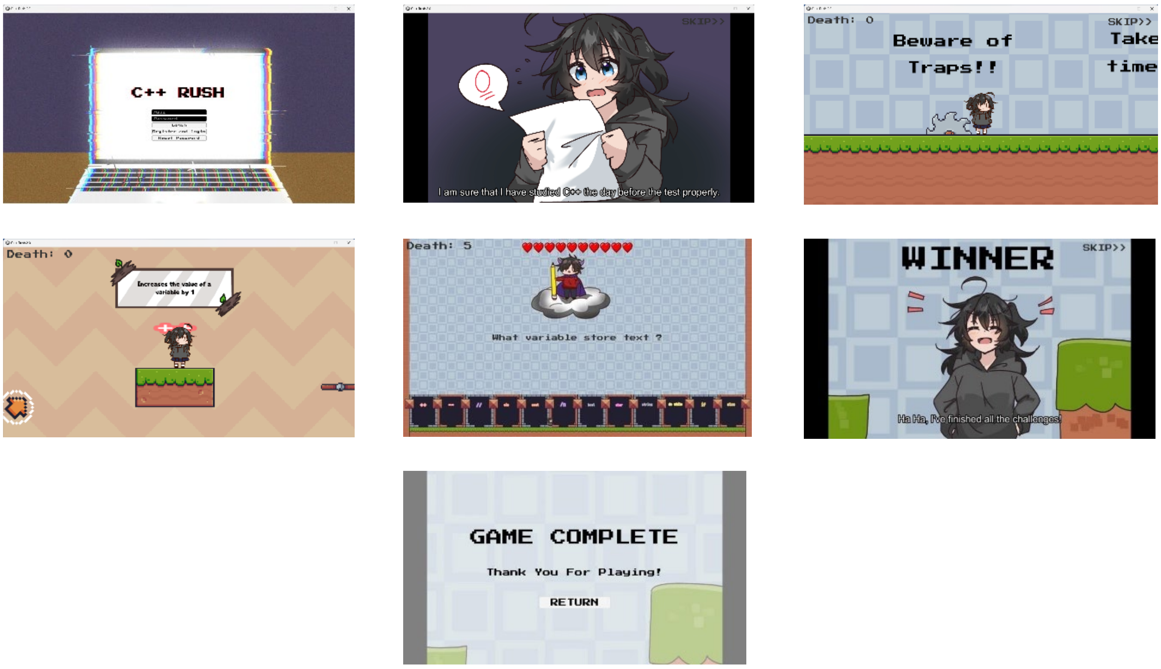

Phases
Activities
Outcome
Requirement Analysis
1. Reading articles about game engine
2. Conducting literature review regarding gamification and related information
3. Collecting data
1. Identified what game engine to be used for the Interactive Game in Learning Computer Language for Novice
2. Identified the gamification element that will be used and any other related information
3. Clear problem statement, objectives, scopes and significant of the project.
System Design
1. Logical design (flowchart and use case diagram)
2. Physical design
3. Architecture design
1. Flowchart and use case diagram of the system
2. System's interface
3. System architecture
Development
1. Develop Interactive Game in Learning Computer Language for Novice using Unity
2. Identify softwares and hardwares requirement
1. Interactive Game in Learning Computer Language for Novice is produced
Testing
1. System testing
1. Testing case
Deployment
1. System deployment
1. System deployed
Review
1. Documentation
1. Project report

Figure 5: Project methodology based on the Agile Model

The first phase of agile methodology was requirement analysis. The goal of the first phase was e completion. Finding an appropriate game engine for the base of the educational game was the initial goal. Reading articles, reviewing literature, and gathering information about gamification were the tasks involved in this phase. Furthermore, it discussed the issue arise, and the objectives, scopes, and significance of the study, the technology which to be utilized for the interactive game for learning computer language for novices, and the information about gamification elements.

The second phase of the methodology was system design. This phase has accomplished the game's second purpose by creating an interactive Game in Learning Computer Language for Novice using the stated approaches and methodologies. Logical designs, such as flowcharts and use case diagrams, as well as physical and architectural designs were created at this phase. This phase's output would be a flowchart and use case diagram describing the system architecture, interface, and design.

The third phase of the methodology was the development of the game. This phase has achieved the second objective of the game. The activities were conducted by developing an Interactive Game for Learning Computer Language for Novice. This phase resulted in the creation of an Interactive Game in Learning Computer Language for Novice. C++ Rush which was constructed using Unity software and stored in the PlayFab platform. The game involved walking, jumping, and avoiding obstacles while learning the C++ syntax. The game has its uniqueness, especially in terms of partaking in challenges while learning the programming language. Instead of learning manually and conventionally, learners may enjoy the game and gain knowledge at the same time. Both methods offer a good learning style for novice programmers and could trigger a passion for learning C++. Screenshots of the game are illustrated in Figure 6.



I am sure that I have studied C++ the day before the test properly.

Increases the value of a variable by 1

What variable store text ?

Ha Ha, I've finished all the challenges!

Figure 6: C++ Rush screenshots

The fourth phase of the methodology was game testing. This phase has achieved the game's third objective, developing an Interactive Game for Learning Computer Language for Novice. The activity of this phase was game testing, and the outcome of the phase was the test case.

The fifth phase of the methodology was deployment. This phase has achieved the game's third objective, developing an Interactive Game for Learning Computer Language for Novice. For the activities, the game was deployed. While deploying the game, the game was tested to see whether it can be successfully deployed. The outcome of the phase the game was successfully deployed. When a game is deployed, the capability is transferred to the ultimate end-user, namely teachers and lecturers.

The sixth phase of the methodology was the game review. This phase has achieved the game's third objective, developing an Interactive Game for Learning Computer Language for Novice. The activity of the phase was documentation of the game and the project report's outcome. Here, the technical documentation would include all written materials and documents related to the entire project's procedure. Ensuring that the researcher and any other relevant parties are moving in the same direction to complete the project's objectives is the principal objective of excellent documentation.

RESULT AND DISCUSSION

Usability is a critical aspect of software systems, and a poor user experience can significantly impact user preferences in favor of alternative software options. Usability testing is a valuable practice for enhancing the user experience, but it often presents challenges, especially in obtaining the active participation of real users (Dias & Paiva, 2017). The evaluation of usability is typically based on criteria such as learnability, efficiency, memorability, error rate, and customer satisfaction (Retnani et al., 2017).

The specific usability testing conducted for C++ Rush involved a carefully selected group of 30 respondents, providing a diverse sample that offered valuable insights into the game's effectiveness across different demographic segments. Most participants, comprising 90% of the total, fell within the 18 to 25 age group, while the remaining 10% belonged to the 26 to 35 age range. This intentional diversification of age groups aimed to capture a broad spectrum of user perspectives, preferences, and levels of familiarity with programming and gamification. The decision to include participants predominantly from the 18 to 25 age group may reflect the target audience for C++ Rush, as younger individuals often form a significant portion of the population interested in learning programming languages. However, the inclusion of participants aged 26 to 35 adds a layer of demographic diversity, considering the varying experiences and expectations that individuals from different age brackets may bring to a learning platform. The feedback obtained from this diverse group of participants serves multiple purposes in the evaluation of C++ Rush's usability. Firstly, it helps assess the game's accessibility and appeal to individuals with varying levels of exposure to programming concepts and gamified learning environments. By including participants with different age ranges, the testing provides a more holistic understanding of how well C++ Rush caters to the preferences and learning styles of a broad audience.

The substantial 83.3% positive response, as reflected in Figure 7, underscores a key aspect of C++ Rush's success: the game's enjoyability. This high level of user satisfaction is indicative of the game's engaging design, emphasizing its crucial role in sustaining user interest and motivation throughout the learning experience. The high percentage of respondents expressing enjoyment suggests that C++ Rush has successfully integrated engaging design elements, fostering an environment that not only educates but captivates users. The game's ability to sustain user interest and motivation is pivotal for achieving its educational objectives, as it ensures that learners remain actively engaged and motivated to explore the intricacies of C++ programming. This positive response sets the foundation for continued success and impact in the realm of gamified programming education.

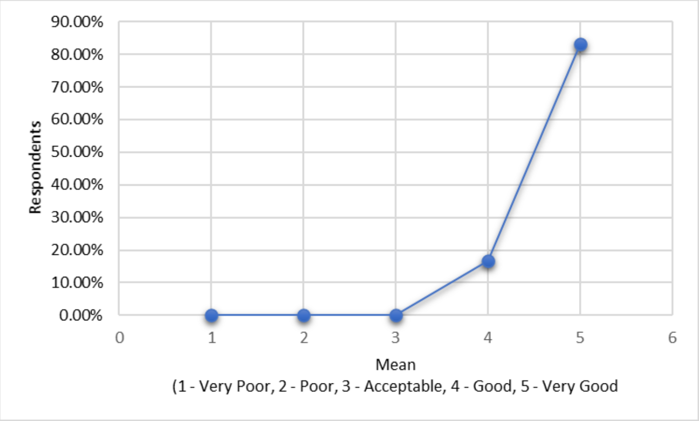


Figure 7: Respondents' feedback about their enjoyment of playing C++ Rush

The data presented in Figure 8 provides compelling evidence of the respondents' overwhelmingly positive perception of gamification as an effective learning method within C++ Rush. The impressive 93.3% agreement among respondents signifies a strong consensus on the efficacy of incorporating gamified elements into the learning experience. This level of support underscores the value of gamification in making the learning process dynamic and engaging, ultimately contributing to heightened motivation and enhanced information retention. The positive responses emphasized not only the perceived effectiveness of gamification within C++ Rush but also the broader recognition of interactive and engaging learning experiences. The agreement among respondents regarding the entertainment value of games further supports the idea that incorporating gamified elements into educational platforms like C++ Rush can be a powerful strategy for making learning enjoyable, motivating, and effective. This positive perception bodes well for the continued success and impact of gamified learning approaches in programming education.

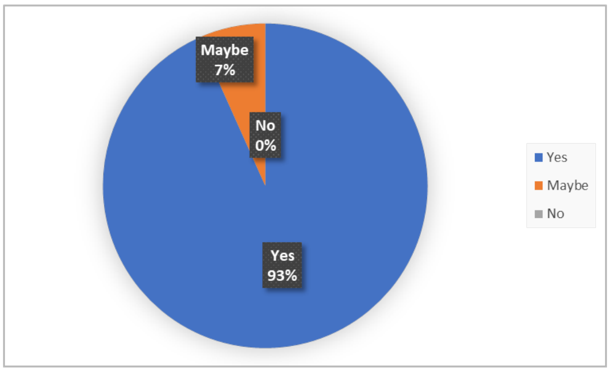


Figure 8: Responds regarding the effectiveness of gamification in learning as implemented in C++ Rush

The findings presented in Figure 9 offer valuable insights into the learning outcomes of the respondents engaged with C++ Rush. The data reveals a notable 66.7% expressing extreme satisfaction with the amount they learned about C++ syntax while playing the game, followed by 23.3% who were satisfied, and 10.0% somewhat satisfied. These responses collectively indicate that C++ Rush has been successful in effectively facilitating the learning of C++ syntax, aligning closely with its educational objectives. The gamified approach, with its focus on hands-on practice, practical application, and adaptability to different skill levels, has resulted in a positive and varied learning experience for users. The high recommendation rate further solidifies the notion that the game has not only met but exceeded user expectations, contributing to the overall success of C++ Rush as an effective educational tool in the realm of programming. Continued feedback and iterative improvements based on user experiences will likely further enhance the platform's educational impact.

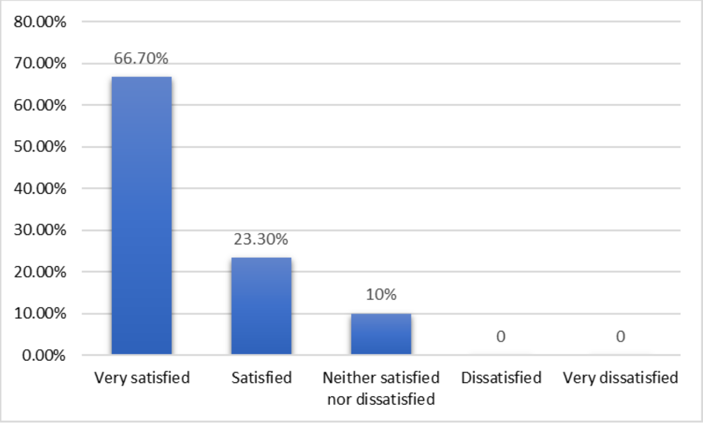


Figure 9: Respondents' feedback on how much they learned from this game

Furthermore, the unanimous agreement (100%) among respondents that they would recommend the game to novice programmers reinforces the confidence and satisfaction they have in C++ Rush. This endorsement suggests that users see value in the game as a learning tool and are motivated to share it with others in the programming community. The affirmative responses from 93.3% of respondents and the additional 6.7% leaning towards agreement on the effectiveness of gamification as a learning method reiterate its positive impact. The dynamic and engaging nature of gamification in C++ Rush appears to have successfully enhanced the learning process, leading to increased motivation and improved information retention.

In conclusion, the findings from the usability testing affirm that C++ Rush has not only succeeded in providing an enjoyable gaming experience but has also effectively leveraged gamification to enhance the learning of C++ syntax. The positive feedback, recommendation rates, and agreement on the efficacy of gamification collectively validate the potential of C++ Rush as an effective and engaging educational tool for novice programmers. Continued user feedback and iterative testing will further contribute to refining and optimizing the usability of C++ Rush for an even broader audience.

CONCLUSION

The study's two-fold objectives, involving the design and development of the interactive educational game C++ Rush, as well as the evaluation of its usability, have yielded promising results. C++ Rush has demonstrated its efficacy as a conducive platform for self-paced learning, offering an engaging and alternative methodology for programming and computer education. This gamified learning model holds significant potential for academicians and researchers seeking innovative tools to aid in the design and development of educational games. Notably, it serves as a valuable resource for elementary students, helping them grasp the fundamental syntax and structure of a programming language interactively and enjoyably. Practically, C++ Rush serves a dual purpose as both an educational and gaming tool. Its versatility makes it an asset in the realm of computer education and learning, especially within the context of children's programming studies. The game's interactive nature not only captures students' interest but also facilitates a more intuitive understanding of complex programming concepts. Moving forward, the development and enhancement of C++ Rush can be steered by actively engaging with a diverse group of respondents for testing and feedback. This approach ensures that the learning models embedded within the game are not only effective but also inclusive. By involving individuals with varying levels of familiarity with both gamification techniques and programming languages, the platform can be fine-tuned to cater to a broad audience, encompassing beginners, intermediate learners, and experienced developers. The recommendation to continue advancing this research is particularly noteworthy. By doing so, C++ Rush can evolve to become even more useful, appealing, valuable, and applicable in the future. Ongoing research and development efforts should focus on refining the game's features, incorporating cutting-edge educational methodologies, and adapting to the evolving landscape of technology and education. In conclusion, C++ Rush stands as a promising tool in the field of gamified learning for programming, with the potential to reshape how students, especially children, approach computer education. The collaborative efforts of educators, researchers, and a diverse group of users will play a pivotal role in elevating C++ Rush to new heights and ensuring its sustained impact on the educational standard of the community.

ACKNOWLEDGEMENTS

Thank you, especially, to Jurnal Intelek for the opportunity to publish, Department of Computer Science and Mathematics, Universiti Teknologi MARA Pulau Pinang, and College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA Cawangan Perlis.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHORS' CONTRIBUTION

The prototype was designed and planned by Shaiful Nizam, M. S. H. He carried out the experiments, prepared the data, and helped analyze the findings. A.F. Rosmani was the main author of the manuscript. Each author contributed ideas that helped refine the study, the analysis, and the manuscript.

CONFLICT OF INTEREST DECLARATION

We certify that the article is the Authors' and Co-Authors' original work. The article has not received prior publication and is not under consideration for publication elsewhere. This research/manuscript has not been submitted for publication nor has it been published in whole or in part elsewhere. We testify to the fact that all Authors have contributed significantly to the work, validity, and legitimacy of the data and its interpretation for submission to Jurnal Intelek.

REFERENCES

Alsawaier, R. S. (2018). The effect of gamification on motivation and engagement. *International Journal of Information and Learning Technology, 35*(1), 56–79.

Anam Fathoni, A. F. C., & Delima, D. (2016). Gamification of learning kanji with "Musou Roman" game. *2016 1st International Conference on Game, Game Art, and Gamification (ICGGAG)*. https://doi.org/10.1109/icggag.2016.8052664

Baharum, A., Moung, E., Ismail, I., Ismail, R., Majalin, M., & Mat. (2020). A preliminary study of difficulties in learning java programming for secondary school. *International Journal of Advanced Trends in Computer Science and Engineering, 9*, 302–306.

Borna, K., & Rad, H. M. (2018). Serious games in computer science learning goals. *2018 2nd National and 1st International Digital Games Research Conference: Trends, Technologies, and Applications (DGRC).*

Dias, F., & Paiva, A. C. R. (2017). Pattern-based usability testing. *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. https://doi.org/10.1109/icstw.2017.65

Grepon, Benzar Glen, Baran, N., Vincent, Martinez, A., & Lacsa, M. (2021). Designing and implementing e-School systems: An information systems approach to school management of a community college in northern Mindanao, Philippines.

Maiga, J. (2019). Gamification for teaching and learning java programming for beginner students - A review. *Journal of Computers, 14*(9), 590–595.

Retnani, W. E. Y., Prasetyo, B., Prayogi, Y. P., Nizar, M. A., & Abdul, R. M. (2017). Usability testing to evaluate the library's academic website. *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*.

Reyssier, S., Hallifax, S., Serna, A., Marty, J.-C., Simonian, S., & Lavoue, E. (2022). The impact of game elements on learner motivation: Influence of initial motivation and player profile. *IEEE Transactions on Learning Technologies, 15*(1), 42–54. https://doi.org/10.1109/tlt.2022.3153239

Rosmani, A. F., & Zakaria, M. H. (2018). Asmaul husna learning through gamifications and adaptation of signalling principle. *Journal of Physics: Conference Series*, 1019, 012080. https://doi.org/10.1088/1742-6596/1019/1/012080

Sailer, M., Hense, J. U., Mayr, S. K., & Mandl, H. (2017). How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior, 69*(69), 371–380.

Seaton, J., Graf, S., Chang, M., & Farhmand, A. (2018). Incorporating learning analytics in an educational game to provide players with information about how to improve their performance. *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*. https://doi.org/10.1109/icalt.2018.00121

Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74, 14–31.

Shaleynikov, A. (2017). *Top 10 programming languages in 2017 - DZone Web Dev*. Dzone.com. https://dzone.com/articles/top-10-programming- languages-in-2017

SoloLearn: Learn to Code. (2019). *SoloLearn*. https://www.sololearn.com/

Wan, G. C., Gao, C., & Tong, M. S. (2017). An efficient method for changing undergraduate students' addiction to the computer games into the interest of learning. *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. https://doi.org/10.1109/tale.2017.8252324

Yu He. (2009). The course choice between C language and C++ language. *2009 4th International Conference on Computer Science & Education*. https://doi.org/10.1109/iccse.2009.5228304

Zamin, N., Ab Rahim, H., Savita, K. S., Bhattacharyya, E., Zaffar, M., & Katijah Mohd Jamil, S. N. (2018). *Learning block programming using scratch among school children in Malaysia and Australia: An exploratory study*. IEEE Xplore.