Jurnal Paedagogy:
Jurnal Penelitian dan Pengembangan Pendidikan
https://e-journal.undikma.ac.id/index.php/pedagogy/index
Email:paedagogy@undikma.ac.id

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

Development of Serious Games as A Programming Learning Platform for Informatics Students

Noven Indra Prasetya*, Shofiya Syidada, Lestari Retnawati
Department of informatics, Faculty of Engineering,
Universitas Wijaya Kusuma Surabaya, Indonesia.
*Corresponding Author. Email: noven@uwks.ac.id

Abstract: This study aims to develop DolananCoding, a programming learning platform based on serious games designed for informatics students. The research employed the waterfall software development methodology, encompassing requirements analysis, system design, implementation, testing, and dissemination. The study participants consisted of second- and fourth-semester students from the Department of Informatics at Universitas Wijaya Kusuma Surabaya, totaling 264 students. The research instrument used is the end-user computing satisfaction (EUCS) questionnaire with a 5-point Likert scale to evaluate user satisfaction with the platform. Data analysis was conducted using descriptive statistics to assess user satisfaction across five dimensions: accuracy, ease of use, content, timeliness and format. The results indicated that the DolananCoding platform was validated and has received positive responses from students, achieving a user satisfaction score of 4.3 out of 5.0. This platform is expected to enhance students' motivation and engagement in programming learning while equipping them with the programming skills required to compete in the professional job market.

Article History
Received: 12-09-2024
Revised: 18-11-2024
Accepted: 20-12-2024
Published: 21-01-2025

Key Words:
Interactive Module; Android; Salt Hydrolysis; High Order Thinking Skills.

How to Cite: Prasetya, N., Syidada, S., & Retnawati, L. (2025). Development of Serious Games as A Programming Learning Platform for Informatics Students. *Jurnal Paedagogy, 12*(1), 12-22. doi:https://doi.org/10.33394/jp.v12i1.12915

https://doi.org/10.33394/jp.v12i1.12915

Introduction

Programming is a core skill that must be mastered by informatics students and those in similar fields of study. It serves as a compulsory course that students must complete to advance to higher-level courses that require programming as a prerequisite. Through learning programming, students develop problem-solving abilities and the capacity to interact effectively with computers (Ekohariadi et al., 2018). Consequently, upon graduation, students are expected to possess skills in system design, computer programming, and user requirements analysis (Rahman & Watanobe, 2023). Furthermore, the growing demand for graduates with strong programming skills in the business and industrial sectors (Vaca-Cárdenas et al., 2015) necessitates that students not only acquire programming abilities but also demonstrate high programming competency (Topalli & Cagiltay, 2018) to compete successfully with other graduates in securing employment opportunities.

Despite its importance, many students face difficulties in understanding programming concepts and syntax when learning it for the first time (Topalli & Cagiltay, 2018). Cohen's research further highlights that almost all first-year students have no prior programming experience (Thorat & Kshirsagar, 2021). According to (Selby, 2015), the initial challenges in learning programming stem from a lack of understanding in reading, tracing, and writing code, particularly with text-based programming languages (Piwek et al., 2019). As a result, many students struggle to complete coursework or final assignments related to programming,

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

even though they have successfully passed prior programming courses (Sophan & Kurniawati, 2018).

The challenges in learning programming are influenced by several factors, including students' limited understanding of programming instructions or syntax (Salvador-Ullauri et al., 2020). This often requires students to engage in continuous practice, progressively advancing their coding skills (Pradana et al., 2023). Additionally, low student motivation is a significant factor contributing to difficulties in learning programming (Jones, 2020). A survey conducted among informatics students at Universitas Wijaya Kusuma Surabaya revealed that approximately 76% of 100 students expressed less interest in programming courses compared to other subjects. Unsurprisingly, a separate study reported that the average programming proficiency score among surveyed students was only 22.89 on a scale of 110 (Ma et al., 2011).

To foster motivation for learning programming among students, a different approach is needed—one that is not only engaging but also assists educators in teaching programming more effectively and enjoyably, such as through games (Caserman et al., 2020). While games are often perceived as a hobby that can detract from productivity due to their addictive nature (Zhao et al., 2021), recent research in game-based learning has introduced the concept of serious games (SG) (Zhao et al., 2022). SG are designed with a serious purpose beyond mere entertainment, often developed to address educational or other practical objectives (Abt, 1987). Unlike traditional games aimed solely at providing enjoyment, SG are utilized both as a teaching tool by educators and as a medium for learners to acquire information or knowledge (Putra et al., 2022). SG create an engaging learning environment aligned with 21st-century skills, such as creativity, problem-solving, and collaboration (Connolly et al., 2012).

This study develops a serious game-based learning platform called DolananCoding, an interactive learning tool for C++ programming courses designed for informatics students. The novelty of this research lies in several key aspects: the serious game-based learning approach, which integrates game elements such as live coding, leaderboards, and immersiveness to enhance students' motivation and engagement in learning programming; the live coding feature, which allows students to write code directly within the game environment, offering a more interactive and hands-on learning experience; leaderboards and experience points, which foster a healthy competitive atmosphere and encourage students to practice more diligently; and the concept of immersiveness, enabling students to acquire knowledge through experiential learning during gameplay. By incorporating live coding features, leaderboards, and the concept of immersiveness, DolananCoding aims to boost students' engagement and motivation in learning C++ programming, ultimately leading to improved learning outcomes.

Research Method

This study employed the waterfall software development methodology (Abba et al., 2019), consisting of requirements analysis, system design, implementation, testing, and publication, as illustrated in Figure 1. The process began with requirements analysis, adapting game principles for programming learning (Almeida & Simoes, 2019) while considering student characteristics. System design encompasses system architecture, database design, and user interface design. The subsequent step involved implementing the platform using programming languages aligned with the system design and predefined technical specifications. These specifications required users to access the platform via a computer or

laptop browser, rather than tablets or smartphones, to ensure the live coding feature functions optimally. The testing phase ensures the platform meets user specifications and requirements, including functionality testing and user satisfaction evaluation. Functionality testing examines the platform's core features, while user satisfaction is assessed through responses from 10-15 participants, evaluating five dimensions using an adapted end-user computing satisfaction (EUCS) instrument (Purwanto & Hedin, 2020). The publication phase involved distributing the platform to students by hosting it and providing a unique URL or domain to ensure easy and widespread access anytime, anywhere.

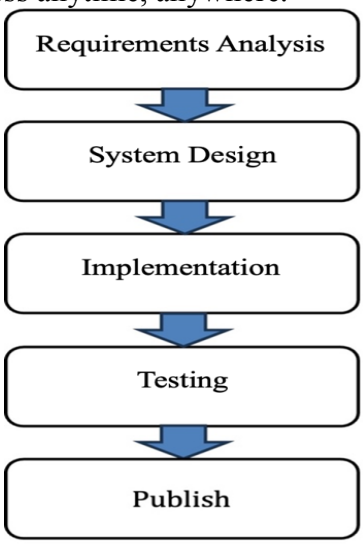

Figure 1. Waterfall Research Method

The research subjects consisted of second- and fourth-semester students from the Informatics program at Universitas Wijaya Kusuma Surabaya, totaling 264 participants. This study involved a large group of students to evaluate their satisfaction with the DolananCoding platform. During the observation period, students explored the platform for five weeks in a online and offline format. Over these, participants explored the platform's features and completed various challenges and exercises. Students completed the end-user computing satisfaction (EUCS) instrument using a 5-point Likert scale to evaluate their satisfaction with the platform during the fifth week. The EUCS, as developed by (Purwanto & Hedin, 2020), focuses on five dimensions: content, format, ease of use, timeliness, and accuracy. Additionally, 12 questions were designed for the EUCS study, which were adopted and used as the basis for developing the questionnaire. Each question was translated into Indonesian to help participants better understand the context.

The data analysis technique employed in this study is descriptive analysis. This method is used to measure user satisfaction based on five key dimensions: accuracy, ease of use, content, timeliness and format. The instrument utilized was the end-user computing satisfaction (EUCS) questionnaire with a 5-point Likert scale. The questionnaire was completed by 276 students after using the DolananCoding platform for five weeks. The data collected from the questionnaire were then analyzed to provide an overview of user satisfaction levels with the DolananCoding platform. The descriptive analysis enables researchers to understand how users evaluate various aspects of the platform and to identify areas that may require further improvement. The analysis results indicated that the DolananCoding platform received positive feedback from students, with a user satisfaction

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

score of 4.3 out of 5.0. This finding demonstrated that the platform was valid and ready to be integrated into programming learning.

Results and Discussion

The results and discussion section presents an explanation of the research findings based on the stages of requirements analysis, system design, implementation, testing, and publication.

Requirements Analysis

The development of a serious game-based programming learning platform resulted in a web-based platform to support C++ programming learning. The platform is designed as an interactive game to enhance student engagement and motivation in structured programming courses. During its use, players select a character as their avatar to complete missions within the game. Each mission and challenge requires players to write programming code through live coding. Players are provided with specific instructions and tasks that must be solved using programming logic. The selection of game elements in this platform is based on three fundamental principles of gameplay: engagement, autonomy & a safe environment, and progression, as illustrated in Table 1.

Table 1. Basic Principles of the Game

Fundamental Principle	Contextual Principle	System Implementation
Engagement	Challenge	Players are presented with problems or cases that must be solved using programming.
	Material	The platform includes features that assist players in applying the concepts they have learned by writing code. By following the provided instructions, players receive immediate feedback. Hints are available to support players who encounter difficulties while completing tasks.
	Character	Each player has an avatar they can control to navigate through the game.
	Exercise	Players can access platform content, review it, and practice the material they have learned through interactive modules.
Autonomy & Safe Environment	Limited Control	Players have autonomy, freedom in their learning process. However, since programming is a subject that builds upon prior knowledge, players are required to complete the game materials sequentially.
Progression	Point	Players earn points upon completing the material for each level.
	Achievement	The platform provides badges, levels, methods for earning badges, and user statistics.

System Design

The system design phase of DolananCoding is divided into three components: system architecture design, database design, and user interface design.

System Architecture

The system architecture was developed based on the results of the requirements analysis, and Figure 2 illustrates the architecture design of DolananCoding. The client side comprises several components grouped into UI/UX and the game engine. The UI/UX

Jurnal Paedagogy:
Jurnal Penelitian dan Pengembangan Pendidikan
https://e-journal.undikma.ac.id/index.php/pedagogy/index
Email:paedagogy@undikma.ac.id

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

components include: (1) Game Interface: Interactive graphics, animations, and visual elements to engage players or students; (2) Code Editor: An integrated development environment (IDE) where players write and test their code; (3) Progress Tracker: Displays players' progress and achievements. The game engine components include: (1) Rendering Engine: Manages graphics and animations; (2) Physics Engine: Simulates real-world physical environments required for the game; (3) Input Handler: Processes player inputs from the keyboard.

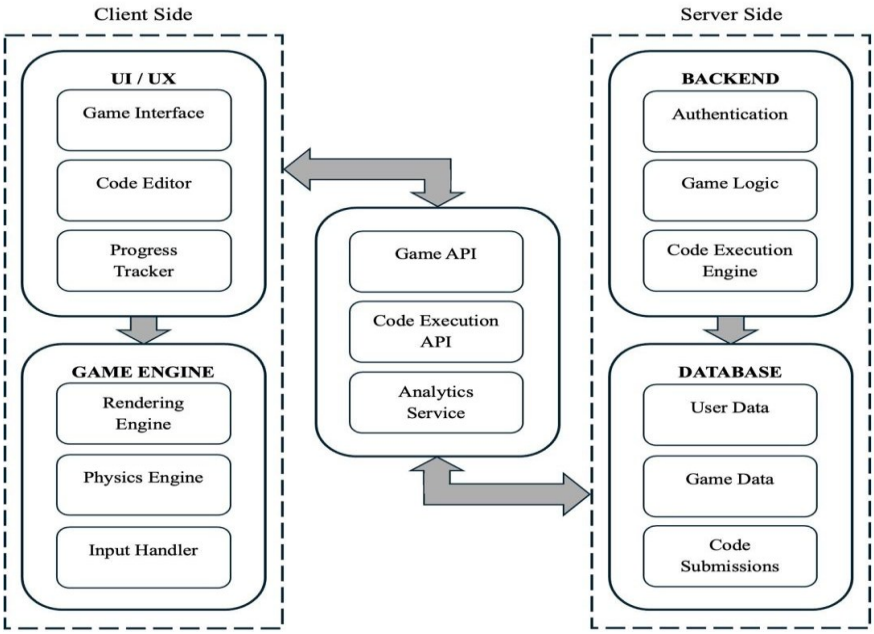

Figure 2. System Architecture

Several components on the server side are categorized into backend and database. The backend components include: (1) Authentication: Manages player login, registration, and access permissions; (2) Game Logic: Controls the rules, scenarios, and progression of the game; (3) Code Execution Engine: Executes the code written by players within a secure sandbox environment and returns the results. The database components include: (1) User Data: Stores player profiles, progress, achievements, and settings; (2) Game Data: Stores levels, challenges, assets, and game configurations; (3) Code Submissions: Archives player-submitted code for future reference and analysis.

Between the server and client, several components are grouped under APIs and services, which include: (1) Game APIs: Serve as the interface for client-server communication, handling game logic and player interactions; (2) Code Execution API: A service for running and testing code written in the C++ programming language; (3) Analytics Service: Collects and analyzes data on player interactions, progress, and performance.

The system flow outlines the sequence of operations and interactions within the platform, detailing how users or players navigate through various components and processes. This flow is divided into four categories: user registration and login, game access and play, code execution and feedback, and progress tracking and analysis. User Registration and Login Flow: (1) Users or players register and log in to the platform through the user interface; (2) The authentication service verifies credentials and authorizes user access. Game Access and Play Flow: (1) Players select a programming challenge or level from the game interface; (2) The game logic server retrieves relevant game data and displays the challenge; (3) Players

write and submit code via the code editor. Code Execution and Feedback Flow: (1) Submitted code is sent to the code execution engine via the API; (2) The code is executed in a sandbox environment; (3) The results and feedback are returned to the player. Progress Tracking and Analysis Flow: (1) Player progress and performance data are stored in the database; (2) The analytics service processes this data to provide insights and personalized recommendations.

Database

The database is utilized to dynamically store all DolananCoding data, including tables for points, badges, and challenges. The points table is designed to store all points earned by players during gameplay. This table includes the following fields: (1) player_id: Stores the player's unique identification; (2) experience_point: Records the total amount of material the player has completed; (3) challenge_score: Tracks the total score the player has accumulated from successfully completed challenges.

Table 2. Point

Field	Data Type	Length	Information
point_id	int	20	PK
player_id	int	20	
experience_point	int	11	
challenge_score	int	11	
created_at	time		
updated_at	time		

The badge table includes the following fields: (1) badge_id: Stores the unique identification of badges earned by each player; (2) player_id: Records the unique identification of players who have earned badges during gameplay.

Table 3. Badge

Field	Data Type	Length	Information
badge_id	int	11	PK
player_id	int	20	
created_at	time		
updated_at	time		

The challenge table is designed to store data on challenges completed by players and includes the following fields: (1) challenge_id: Stores the unique identification of challenges completed by each player; (2) user_id: Records the unique identification of players who have completed the challenges; (3) score: Stores the score achieved for each completed challenge; (4) code: Stores the programming code written by players to complete the challenges.

Table 4. Challenge

Field	Data Type	Length	Information
challenge_id	int	11	PK
player_id	int	20	
score	int	5	
code	text		
created_at	time		
updated_at	time		

Each table includes the following fields: (1) created_at: Records the timestamp when a player first earns points or completes a related activity; updated_at: Records the timestamp when points or other related data are subsequently updated.

Interface Design

Figure 3 illustrates the interface design for the map and character selection page in DolananCoding. The map page appears after players successfully log in to the platform.

Jurnal Paedagogy:
Jurnal Penelitian dan Pengembangan Pendidikan
https://e-journal.undikma.ac.id/index.php/pedagogy/index
Email:paedagogy@undikma.ac.id

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

Players can select a map to start the game. However, for first-time play, players are only allowed to choose the available map. Other maps will unlock as players complete all game levels on the previous map. Each map consists of ten game levels, with each level containing C++ programming material or cases that must be solved through live coding.

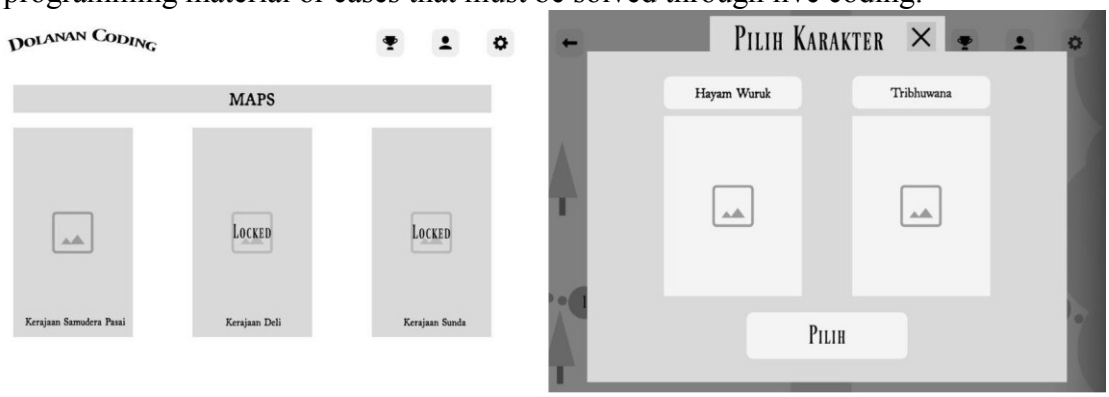

Figure 3. Interface Design

The character selection page allows players to choose an avatar to represent them in navigating the game. The platform offers two characters: Hayam Wuruk, representing a male character, and Tribhuwana, representing a female character. Both character names are derived from the names of a king and queen of the Majapahit Kingdom in Indonesia.

Implementation

The implementation phase aims to transform the concepts and designs based on the architecture, database, and user interface into a functional product that users can utilize. DolananCoding was developed using Unity as the game engine, with C#Script and JavaScript used for scripting the game logic. All game assets, including characters, textures, animations, and audio, were imported into Unity and integrated with game components to create a more realistic gaming environment. Controls, such as movement, attacks, and interactions, were configured using computer input devices to make the characters and objects playable. Game logic scripting was also implemented to define rules, gameplay mechanics, and game systems such as level progression, scoring, inventory, and save systems. Target platform selection was performed before the build process to align the game's distribution with the research objective—deployment on a web platform. Table 5 outlines the software specifications used during the implementation phase.

Table 5. Software Specifications

Software Name	Specification
Operating System	Windows 11 64-bit
Game Engine	Unity
Database	MySQL
Programming	C#Script, JavaScript
Web Component	CodeMirror, WebGL
Code Editor	Visual Studio Code

The implementation resulted in a web-based platform, DolananCoding, featuring a main page, login page, account creation page, map selection, character selection, level selection, and a gameplay area, all aligned with the interface design described in the previous sections.

Testing

This study validated the platform through unit testing, system testing, and user testing. The approach began with unit testing, which modeled pseudocode as a flowgraph to evaluate the basic paths. Unit testing aimed to determine the platform's complexity and its independent

Jurnal Paedagogy:
Jurnal Penelitian dan Pengembangan Pendidikan
https://e-journal.undikma.ac.id/index.php/pedagogy/index
Email:paedagogy@undikma.ac.id

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

paths. The second stage, system testing, ensured that the system's functional requirements were met using the black-box testing method. The final stage, user testing, was conducted with a group of students to evaluate the platform's content, format, ease of use, timeliness, and accuracy.

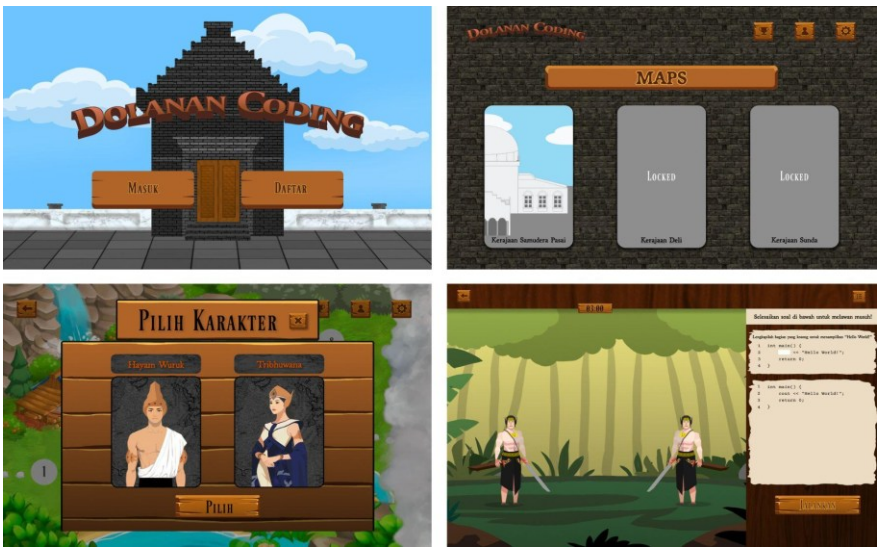

Figure 4. UI DolananCoding

The results of the unit testing are presented in Table 6, covering three main modules: content, grading, and leaderboard. These modules were randomly selected from methods related to game elements. The testing report indicates that the system passed the tests, showing no detected vulnerabilities and demonstrating predictable usability for students (Mishra et al., 2007).

Table 6. Unit Testing

Module	R	N	E	P	VG	Results
Material	4	9	10	2	6	Valid
Grade	3	9	10	3	5	Valid
Leaderboard	3	6	7	3	5	Valid

The results of the system testing, as shown in Table 7, were conducted using the black-box testing method. This validation aims to ensure that all functional requirements were met as expected. The tested system components included content delivery, experience point calculations, and progress tracking.

Table 7. System Testing

Module	Expectation	Results
Material	The platform is capable of displaying C++ programming materials.	Valid
Experience Points	The platform is capable of calculating experience points based on the difficulty level of the tasks being completed.	Valid
Progress	The system is capable of calculating player progress in the form of percentages.	Valid

Additionally, user testing was conducted by distributing questionnaires to 276 students. The class explored the platform over five weeks to complete programming materials and challenges. Table 8 presents the results of user satisfaction, measured using variable-based questionnaire instruments and analyzed through descriptive statistics. These findings align with a previous study by (Pradana et al., 2023), which reported user satisfaction scores above 4.00, indicating that the platform received positive feedback from users and is ready for implementation and integration into programming learning.

Jurnal Paedagogy:
Jurnal Penelitian dan Pengembangan Pendidikan
https://e-journal.undikma.ac.id/index.php/pedagogy/index
Email:paedagogy@undikma.ac.id

Vol. 12 No. 1 : January 2025
E-ISSN: 2722-4627
pp. 12-22

Table 8. User Testing

ID	Item	N	Min	Max	Mean	Std Dev	Level	Results
A2	Accuracy	0.838	2	5	4.17	0.627	5	Very Strong
C4	Ease of Use	0.779	2	5	4.48	0.646	5	Very Strong
C1	Content	0.829	1	5	4.22	0.645	5	Very Strong
T2	Timeliness	0.731	2	5	4.31	0.505	5	Very Strong
F1	Format	0.669	1	5	4.32	0.502	5	Very Strong

Publish

The publishing process was carried out to distribute the platform to end users, specifically students. Distribution involved hosting the platform on a central hosting service and providing a unique URL or domain. This approach ensures broader accessibility, allowing users to access the platform easily anytime and anywhere based on their needs. The conceptual implications of this study indicate that the use of serious games in programming learning can enhance students' motivation and engagement. This concept supports the theory that game elements can make the learning process more engaging and enjoyable, which, in turn, can improve learning outcomes. The study also reinforces the perspective that interactive and experiential learning approaches can help students better understand programming concepts (Connolly et al., 2012; Zhao et al., 2021).

Practically, the findings of this study provide guidance for educators and educational institutions to adopt serious game-based learning platforms like DolananCoding in their curricula. With features such as live coding, leaderboards, and immersiveness, this platform can serve as an effective tool to enhance students' programming skills. Moreover, it can be integrated into various programming courses to offer a more interactive and enjoyable learning experience.

Conclusion

This study showed that the development of the serious game-based learning platform, DolananCoding, successfully enhanced students' motivation and engagement in learning C++ programming. The platform integrated game elements such as live coding, leaderboards, and immersiveness, which have proven effective in making the learning process more engaging and enjoyable. Testing results demonstrated that DolananCoding was valid and received positive feedback from students, achieving a user satisfaction score of 4.3 out of 5.0. This indicates that the platform is ready for integration into programming learning and can assist students in mastering the programming skills needed to compete in the job market. Thus, DolananCoding presents an innovative solution to address the challenges students often face in learning programming.

Recommendation

Based on the findings of this study, several recommendations can be made for educators, students, and future researchers. For educators, it is recommended to integrate the DolananCoding platform into the curriculum of programming courses to enhance student engagement and motivation. The use of features such as live coding and leaderboards within DolananCoding can encourage students to actively participate in the learning process. Additionally, educators can utilize data from the leaderboard and experience points to provide constructive feedback to students and conduct periodic evaluations to identify areas requiring improvement.

For students, it is recommended to fully utilize all features available in DolananCoding, such as live coding, leaderboards, and immersiveness, to enhance their programming skills. Students can also use this platform for independent practice outside of class hours to improve their understanding and proficiency in programming. Additionally, students can leverage the leaderboard to engage in healthy competition with their peers and collaborate on solving programming challenges.

For future researchers, it is recommended to further develop the DolananCoding platform by incorporating new features that can enhance students' learning experiences, such as adaptive learning modules or integration with augmented reality (AR) technology. Researchers can also conduct further studies to evaluate the platform's effectiveness in improving student learning outcomes compared to conventional teaching methods. Additionally, comparative studies between DolananCoding and other programming learning platforms could be undertaken to identify the strengths and weaknesses of each platform.

Acknowledgment
Acknowledgments are extended to Universitas Wijaya Kusuma Surabaya for providing research funding and support, enabling the successful completion of this study.

References

Abba, S., Wadumi Namkusong, J., Lee, J.-A., & Liz Crespo, M. (2019). Design and Performance Evaluation of a Low-Cost Autonomous Sensor Interface for a Smart IoT-Based Irrigation Monitoring and Control System. Sensors, 19(17), Article 17. https://doi.org/10.3390/s19173643

Abt, C. C. (1987). Serious Games. University Press of America.

Almeida, F., & Simoes, J. (2019). The Role of Serious Games, Gamification and Industry 4.0 Tools in the Education 4.0 Paradigm. Contemporary Educational Technology, 10(2), 120–136. https://doi.org/10.30935/cet.554469

Caserman, P., Hoffmann, K., Müller, P., Schaub, M., Straßburg, K., Wiemeyer, J., Bruder, R., & Göbel, S. (2020). Quality Criteria for Serious Games: Serious Part, Game Part, and Balance. JMIR Serious Games, 8(3), e19037. https://doi.org/10.2196/19037

Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. Computers & Education, 59(2), 661–686. https://doi.org/10.1016/j.compedu.2012.03.004

Ekohariadi, E., Anistyasari, Y., Putra, R., & Kurniawan, I. (2018). Assessing Computational Thinking using Pseudocode Programming Instrument. 134–138. https://doi.org/10.2991/aptekindo-18.2018.30

Jones, B. D. (2020). Motivating and Engaging Students Using Educational Technologies. In M. J. Bishop, E. Boling, J. Elen, & V. Svihla (Eds.), Handbook of Research in Educational Communications and Technology: Learning Design (pp. 9–35). Springer International Publishing. https://doi.org/10.1007/978-3-030-36119-8_2

Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011). Investigating and improving the models of programming concepts held by novice programmers. Computer Science Education, 21(1), 57–80. https://doi.org/10.1080/08993408.2011.554722

Mishra, A., Ercil Cagiltay, N., & Kilic, O. (2007). Software engineering education: Some important dimensions. European Journal of Engineering Education, 32(3), 349–361. https://doi.org/10.1080/03043790701278607

Piwek, P., Wermelinger, M., Laney, R., & Walker, R. (2019). Learning to program: From problems to code. Proceedings of the 3rd Conference on Computing Education Practice, 1–4. https://doi.org/10.1145/3294016.3294024

Pradana, F., Setyosari, P., Ulfa, S., & Hirashima, T. (2023). Development of Gamification-Based E-Learning on Web Design Topic. International Journal of Interactive Mobile Technologies (iJIM), 17(03), Article 03. https://doi.org/10.3991/ijim.v17i03.36957

Purwanto, & Hedin, P. B. D. (2020). Measurement of user satisfaction for web-base academic information system using end-user computing satisfaction method. IOP Conference Series: Materials Science and Engineering, 909(1), 012044. https://doi.org/10.1088/1757-899X/909/1/012044

Putra, D. A. T. B., Adisusilo, A. K., & Prasetya, N. I. (2022). Optimasi Aset dan Karakter Permainan 3D Berbasis Tematik Sekolah Dasar. Journal of Information System,Graphics, Hospitality and Technology, 4(01), 1–6. https://doi.org/10.37823/insight.v4i01.165

Rahman, Md. M., & Watanobe, Y. (2023). ChatGPT for Education and Research: Opportunities, Threats, and Strategies. Applied Sciences, 13(9), 5783.

Salvador-Ullauri, L., Acosta-Vargas, P., Gonzalez, M., & Luján-Mora, S. (2020). Combined Method for Evaluating Accessibility in Serious Games. Applied Sciences, 10(18), 6324. https://doi.org/10.3390/app10186324

Selby, C. C. (2015). Relationships: Computational thinking, pedagogy of programming, and Bloom's Taxonomy. Proceedings of the Workshop in Primary and Secondary Computing Education, 80–87. https://doi.org/10.1145/2818314.2818315

Sophan, M. K., & Kurniawati, A. (2018). Perancangan Aplikasi LEARNING BY DOING INTERAKTIF untuk Mendukung Pembelajaran Bahasa Pemrograman. Jurnal Teknologi Informasi Dan Ilmu Komputer, 5(2), 163–170.

Thorat, S. A., & Kshirsagar, D. P. (2021). Developing Logic Building, Problem Solving, and Debugging Programming Skills Among Students. Journal of Engineering Education Transformations, 34(0), 402–406. https://doi.org/10.16920/jeet/2021/v34i0/157188

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. Computers & Education, 120, 64–74. https://doi.org/10.1016/j.compedu.2018.01.011

Vaca-Cárdenas, L. A., Bertacchini, F., Tavernise, A., Gabriele, L., Valenti, A., Olmedo, D. E., Pantano, P., & Bilotta, E. (2015). Coding with Scratch: The design of an educational setting for Elementary pre-service teachers. 2015 International Conference on Interactive Collaborative Learning (ICL), 1171–1177. https://doi.org/10.1109/ICL.2015.7318200

Zhao, D., Muntean, C. H., Chis, A. E., & Muntean, G.-M. (2021). Learner Attitude, Educational Background, and Gender Influence on Knowledge Gain in a Serious Games-Enhanced Programming Course. IEEE Transactions on Education, 64(3), 308–316. https://doi.org/10.1109/TE.2020.3044174

Zhao, D., Muntean, C. H., Chis, A. E., Rozinaj, G., & Muntean, G.-M. (2022). Game-Based Learning: Enhancing Student Experience, Knowledge Gain, and Usability in Higher Education Programming Courses. IEEE Transactions on Education, 65(4), 502–513. https://doi.org/10.1109/TE.2021.3136914

