Priviet Social Sciences Journal
E-ISSN 2798-6314
P-ISSN 2798-866X
https://doi.org/10.55942/pssj.v5i10.754

Volume 5
Issue 10
October Edition 2025

Development of the 2D mobile game "Highwash Rush" with the theme of avoiding vehicle obstacles using unity with the MDCL method

Charmiyanti Nurkentjana A.*, Yusran Yasir, Indah Fajrin Kamil Falaq, Dwi Wibisono

Teknologi Rekayasa Multimedia, Boash Polytechnic, Bantarsari, Kec. Ranca Bungur, Kab. Bogor, Jawa Barat 16310
e-mail: charmiyantina03@gmail.com

Received 17 July 2025
Revised 22 September 2025
Accepted 28 October 2025

ABSTRACT

Among the most prevalent forms of digital entertainment are two-dimensional (2D) mobile games, which are engaging, readily accessible, and compatible with a wide range of devices. This study sought to design and implement a 2D obstacle-avoidance game using Unity. The development methodology employed is the Multimedia Development Life Cycle (MDLC), comprising six stages: concept, design, material collection, assembly, testing, and distribution. The game, named *Highwash Rush*, was developed with various control features, including gyroscope sensors, swipe gestures, and directional buttons, and is equipped with a progressive scoring system and sound effects to enhance player experience. Black Box and Alpha Testing conducted on several Android devices demonstrated that the game operated stably, responded effectively, and adhered to the original design specifications. The findings suggest that implementing the MDLC method, excluding the distribution stage, in unity-based educational game development is effective for producing engaging and interactive multimedia applications suitable for practical use.

Keywords: 2D Game, Unity, MDLC, Android, Interactive Multimedia.

1. INTRODUCTION

In the past five years, advances in smartphone technology have significantly driven mobile gaming industry growth. Owing to their simple, addictive, and easily accessible gameplay, 2D games have emerged as one of the most favored categories. Speed, accuracy, and ease of control are critical factors in the development of 2D games because they directly influence user engagement (Jitendra et al., 2021). Obstacle-avoidance games have also become increasingly popular because they provide progressive challenges that encourage players to improve their scores and extend the gameplay duration (Poondej & Lerdpornkulrat, 2019).

Various platforms are employed in the field of game development, one of the most widely used being Unity. Unity has become a leading platform, particularly for 2D mobile games, as it offers a comprehensive range of features, including asset management, animation processing, scripting with C#, and cross-platform export capabilities for operating systems such as Android, iOS, and Windows (Ferdinand et al., 2024). Another advantage of Unity lies in its extensive global community support and official documentation, which facilitates the learning process for beginners. According to Prayudha and Chotijah (2024), Unity is the primary choice for both educational and entertainment game development because of its time efficiency and the quality of the resulting products.

Atika (2025) reports that approximately 60% of educational game developers in Indonesia use Unity as their primary platform. This is largely due to its user-friendly interface and compatibility across multiple devices. Such evidence highlights the substantial potential of games to serve as enjoyable and practical alternative educational media, particularly for children and adolescents. Games centered on avoiding vehicular obstacles are considered to provide significant educational benefits, including the enhancement of alertness, reflexes, visual-motor coordination, and risk awareness in traffic environments.

The purpose of this study is to design and implement a 2D mobile game with the theme of avoiding vehicular obstacles using Unity. The game, titled *Highwash Rush*, is specifically targeted at children and adolescents aged 10–18 years. This study adopts the Multimedia Development Life Cycle (MDLC), which consists of six development stages: concept, design, material collection, assembly, testing, and distribution. This method was selected because it provides a systematic framework and enables evaluation at each stage of development, thereby ensuring a higher quality final product (Maulana et al., 2022).

2. THEORETICAL BACKGROUND

The Multimedia Development Life Cycle (MDLC) is a systematic, iterative, and structured approach designed to produce multimedia products that are effective, efficient, and aligned with user needs. This model comprises six main stages: concept, design, material collection, assembly, testing, and distribution. At the concept stage, developers define the project's objectives; during the design stage, they create visual and interactive system layouts; and in the material collection stage, they gather media to be used, such as video, audio, and images. In game development, MDLC facilitates teamwork by providing a clear workflow and enabling iterative revisions at each stage of the process, thereby reducing development errors (Prayudha & Chotijah, 2024).

Another strength of the MDLC method is its capacity to effectively integrate visual, audio, and user-interactive elements. In the context of educational and entertainment game development, MDLC supports the creation of content that is not only visually appealing but also capable of conveying specific messages or values to users. Sari et al. (2024) Games developed with MDLC exhibit higher levels of user interaction quality because design and evaluation are embedded throughout the process from initiation to completion (Binanto, 2010; Aleem et al., 2016; Putra et al., 2023; Aryani et al., 2024). Furthermore, MDLC provides flexibility in the selection of development tools and platforms, whether web-based or mobile-based, such as Android (Saputra & Setiawan, 2024; Maulana et al., 2022). The model is well suited for game-based educational projects, as its structured stages effectively combine educational and entertainment values into a single digital product. Prayudha and Chotijah (2024) further note that the model is particularly favored by novice developers because its clear workflow offers a systematic framework for developing interactive products.

Accordingly, the MDLC not only represents a relevant development approach in multimedia but has also been shown to produce optimal results in mobile-based 2D game projects. By accounting for visual design, interactive control, and overall gameplay logic, the model facilitates the creation of comprehensive user experiences (Prayudha & Chotijah, 2024). For example, the MDLC supports the development of games such as *Highwash Rush* by providing a structured process that enables task allocation, periodic evaluation, and the delivery of a stable and usable final product. Thus, MDLC can be regarded as one of the most effective approaches for multimedia application development, particularly for interactive and educational games that require full cross-media integration (Saputra & Setiawan, 2024; Maulana et al., 2022).

From a methodological perspective, the Multimedia Development Life Cycle (MDLC) has become a widely adopted framework for building interactive multimedia content such as educational games. Comprising six stages—concept, design, material collection, assembly, testing, and distribution—the MDLC enables a systematic and iterative development process. Maulana et al. (2022) argue that MDLC is ideal for digital product development because it facilitates continuous evaluation at each stage. Its application in visual game development has been shown to improve the overall quality of the final product by allowing iterative refinement. Saputra and Setiawan (2024) reinforce this view, highlighting that the MDLC is particularly suitable for educational game development because it integrates visual design, gameplay logic, and user interactivity within a structured framework.

3. METHODOLOGY

3.1. MDLC Development Model

This study employs the Multimedia Development Life Cycle (MDLC) as the primary methodological framework for the development process. The MDLC is an iterative, systematic, and structured approach to multimedia development designed to produce digital products that are both effective and aligned with end-user needs. The method consists of six main stages: (1) concept, (2) design, (3) material collection, (4) assembly, (5) testing, and (6) distribution (Purwanti et al., 2022).

Each stage of the MDLC is designed to address multimedia development comprehensively, ranging from idea formulation to the delivery of a final, user-ready product. This structure also allows for gradual evaluation and revision, thereby ensuring optimal outcomes. In the context of interactive game development, such as *Highwash Rush*, the MDLC model is particularly suitable because it accommodates user interface (UI) design requirements, integration of visual and audio assets, and stepwise application testing prior to final distribution.

The adoption of the MDLC is further supported by prior research, which highlights its effectiveness in producing high-quality multimedia products within efficient development timelines, particularly in the context of educational games and user-interactive applications (Saputra & Setiawan, 2024). Figure 1 illustrates the MDLC workflow, which encompasses six sequential stages, from concept formulation to final product distribution. With such a structured process, development teams can better divide roles and responsibilities at each stage, systematically document project progress, and identify potential technical or design-related obstacles early. Moreover, the model supports consistency across the workflow and ensures integration among phases, ultimately leading to multimedia products that are both functional and suitable for distribution (see Figure 1)

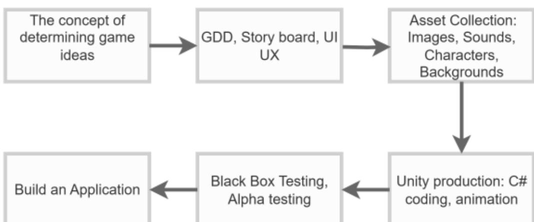


Figure 1. The process of MDLC

3.2. Development Environment

The development of the *Highwash Rush* game was carried out in a work environment that supports the creation of interactive multimedia applications, encompassing both hardware and software components. The selection of tools and software specifications was adjusted to meet the minimum requirements for 2D game development using Unity, while also considering the capacity to handle features such as graphics processing, programming, and testing on an Android device.

The following hardware and software were utilized during the development process (see Table 1 & 2)

Table 1. Game Development Hardware

Device Type	Specification	Function
Laptop	Intel Core i5 Gen 10, RAM 8GB, SSD 512GB, GPU Intel UHD Graphics.	Used to run development software such as Unity, Visual Studio, Figma, and Canva.
Smartphone Testing 1	Realme Note 60, MediaTek Helio G85, RAM 6 GB, Internal Memory 128 GB, Android 14.	Used to test the results of car game builds directly on mobile devices and evaluate performance and user experience.
Smartphone Testing 2	Poco M4 PRO, CPU Helio G96 OctaCore Max2.05GHz, RAM	Used to directly test the game build on mobile devices and to evaluate both performance and user experience

Table 2. Game Development Software

Software Name	Function
Unity	Used as the main engine for creating, managing physics, graphics, and gameplay for car games.
Microsoft Visual Studio	To write and edit C# scripts used in game logic, such as car controls and point systems.
Figma	Designing the user interface (UI), such as the main menu, start button, and score display.
OpenAI	Generating ideas, assets, or programming assistance such as characters or vehicle AI systems.

3.3. Game System Design

Game system design is a crucial stage that forms the foundation for implementing the gameplay logic, control flow, and core functionalities of the application. This stage aims to systematically visualize how the game will operate before proceeding to coding and testing. In the context of *Highwash Rush* development, the design process ensures that all user interactions with the system follow the intended flow and remain intuitive for the players.

System design also plays a vital role in preventing implementation errors, as it provides a comprehensive overview of the gameplay flow, player decision-making, and potential conditions that may occur during the game. The process includes the design of the main menu interface, control buttons, core gameplay sequence, as well as the system's handling of events such as collisions, scoring, and game over. With a well-structured design, developers can more easily identify the critical components of the system and prioritize feature development according to player needs.

In this project, two primary design models were employed: the game system flowchart and the use case diagram. The flowchart illustrates the sequential logical steps that occur during gameplay, beginning from the moment the player launches the game, selects a menu, enters the gameplay session, and continues until the game is over. Meanwhile, the use case diagram maps the interactions between the user (player) and the system, while also defining the essential functionalities that must be available within the game.

3.3.1 Game System Flowchart

The flowchart is used to illustrate the structured sequence of gameplay logic from start to finish. The diagram represents the process beginning with the main menu, the selection of the "Play" button, the progression of the gameplay, score calculation, obstacle collision detection, and finally the game-over condition, along with options for restart or exit. This flow is critical for guiding the scripting process in Unity to ensure that the game operates according to the predetermined design.

By utilizing a flowchart, the development team can clearly understand the sequence of processes within the game and minimize the logical errors during implementation. The diagram also aids in identifying key interaction points between the player and the system, thereby allowing the programming logic to be structured efficiently and responsively to user actions (see Figure 2).

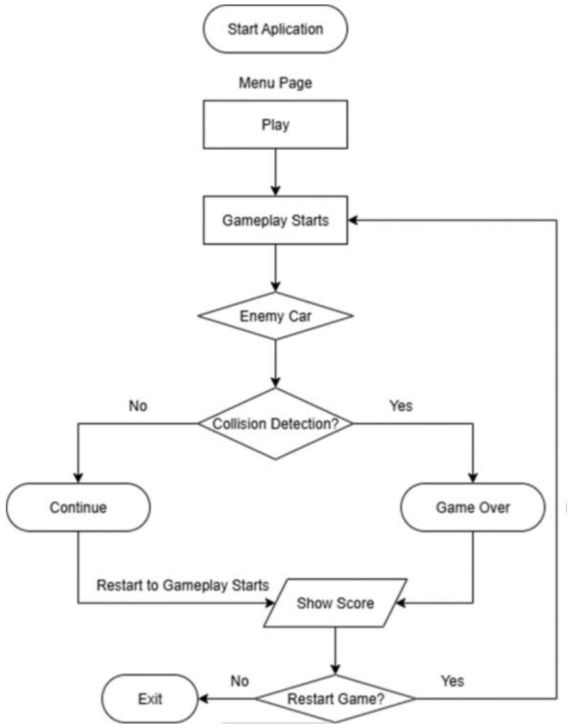


Figure 2. Flowchart System Game

3.3.2 Game System Use Case Diagram

The use case diagram is employed to visually and structurally represent the interactions between the user (player) and the game system. Its purpose is to identify the core features that must be available within the game and to illustrate how players interact with those features. In *Highwash Rush*, players can perform various actions, such as starting the game, adjusting sound settings, controlling character movement, avoiding obstacles, restarting the game after a game-over condition, and exiting the application.

The construction of the use case diagram assists the development team in understanding the functional scope of the system being developed. Moreover, the diagram serves as a critical reference during the implementation stage to ensure that all user requirements are translated into the appropriate features. By providing a clear visualization, the use case diagram enhances team coordination, particularly between interface designers and developers, ensuring a more effective and well-directed development process (see Figure 3).

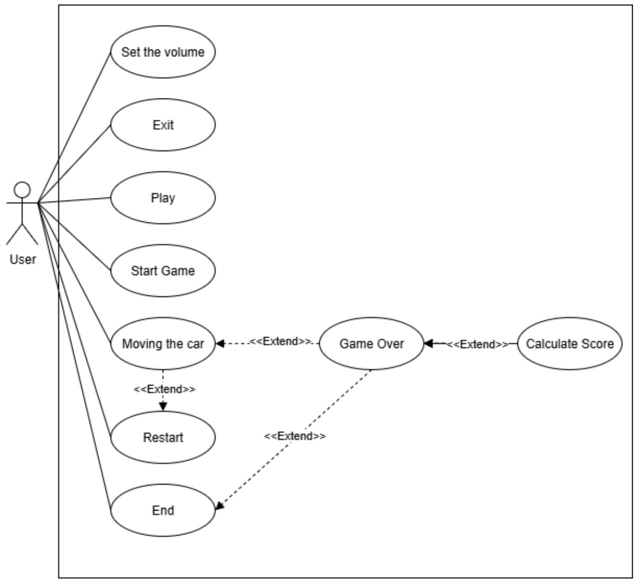


Figure 3. ***Usecase Diagram System Game***

4. RESULT AND DISCUSSION

4.1. Implementation Game

The *Highwash Rush* game was successfully implemented using Unity as the primary game engine and C# as the programming language for scripting the gameplay logic. The implementation process was conducted after completing all system design stages, including user interface design and the collection of visual and audio assets. This stage encompassed several key components of the game, such as the main menu interface, character control system, obstacle generation, scoring system, background music, sound effects, and handling of game over.

At this stage, all previously collected assets were integrated into the Unity. The main character was designed to move left and right using virtual control buttons displayed on the screen. Obstacles in the form of vehicles were generated randomly from the front using the *ObstacleSpawner.cs* script, whereas collisions between the character and vehicles were processed using the *CollisionHandler.cs* script. The core gameplay logic, including scoring and game reset functions, was managed by the *GameManager.cs* script.

The game interface was intentionally designed to be simple and intuitive for players aged 10–18 years. The main menu consisted of a Play button, an Exit button, and the game logo. In the gameplay view, essential elements such as the character, obstacles, score, and control buttons were arranged in an accessible layout. Sound effects, such as vehicle horns and collision sounds, were integrated to enhance the overall playing experience.

The implementation process was conducted incrementally and continuously tested throughout development. Each completed feature was immediately tested to minimize the occurrence of bugs during the final testing stage. The game was developed in portrait mode to ensure comfortable gameplay on Android devices. Once all the features were functioning properly, the game was compiled into an APK format and prepared for subsequent testing

Figure 4. Main Menu Display

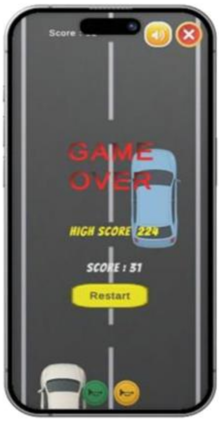
Figure 4 shows the initial interface that appears when the player launches the application. The interface includes a *Play* button to start the game and an *Exit* button to close the application. The menu layout was designed to be simple and intuitive, ensuring ease of use for players of a wide range of age groups.

Picture 5. *Gameplay Display*

Figure 5 illustrates the main gameplay screen. The character is positioned at the bottom of the screen and can be moved left or right using virtual control buttons. Players must avoid obstacles in the form of vehicles approaching from the front of the vehicle. The score is displayed at the top of the screen as an indicator of gameplay progress.

Figure 6 shows the *game-over* condition when the character collides with an obstacle. At this stage, players are given the option to restart the game using the *Restart* button or return to the main menu. This screen marks the end of a gameplay session and offers players the opportunity to try again.

The scoring system is based on the distance traveled by the player's vehicle and the number of other vehicles successfully avoided. For every 10 seconds of gameplay, the player earns 100 points. Both the score and the *High Score* are displayed when the player collides with an obstacle and the game ends. The *High Score* represents the highest score achieved during gameplay.



Gambar 6. ***Game Over Display***

4.2. Testing Results

Testing is a crucial stage in the development of interactive applications to ensure that all features and functions are implemented and operate as intended according to the initial design. In this study, testing was conducted after the completion of all implementation stages of the *Highwash Rush* game, with the primary objectives of evaluating the system stability, functional accuracy, and user comfort during gameplay.

The testing methods applied consisted of two types: Black Box Testing and Alpha Testing. Black Box Testing was performed to examine the functionality of each feature without reviewing the program's source code, whereas Alpha Testing aimed to collect feedback from early users regarding the gameplay experience, including aspects of control, visuals, and overall flow.

4.2.1 Black Box Testing

Black Box Testing was employed to validate the functions within the game without directly inspecting the program code. The main focus of this testing was on inputs and outputs, as well as on how the system responds to player actions. Each feature was evaluated based on its expected performance. Table 3 presents the results of testing the main game functions.

Tabel 3. Blackbox Testing

Test Case	Test Steps	Expected Outcomes	Status
Press the "Play" button	Click the Play button from the main menu.	The game starts, and characters appear in the arena	Succeed
Move the character to the left and right	Swipe the screen, tilt your phone, or press the arrow keys.	The character moves in the appropriate direction.	Succeed
Collision detection	Let the character hit the vehicle.	Game Over appears, final score displayed	Succeed
Restart the game	After Game Over, press the Restart button.	The game starts from the beginning	Succeed
Disable and enable sound	Tap the sound icon in the upper right corner	Sound turned off/on depending on conditions	Succeed
Exit the app	Press the Exit button from the main menu.	Closed the game and returned to the Android home screen	Succeed

4.2.2 Alpha Testing

Alpha Testing was conducted by the internal development team using two Android devices to ensure that the application runs smoothly on target devices with varying specifications. The primary focus of this testing was the application stability, control performance, and compatibility of the user interface design across different screen sizes. Table 4 is a summary of the Alpha Testing evaluation results:

Tabel 4. Alpha Testing

Perangkat	Specifications	Test Results
Realme Note 60	Android 14, RAM 6 GB, MediaTek Helio G85	The game runs well, without lag, and the user interface is visibly perfect.
Poco M4 Pro	Android 11, RAM 8 GB, Helio G96	Responsive performance, good sound quality, accurate sensor control

Based on the results of Alpha Testing, no crashes or major bugs were found during gameplay. All control systems, including swipe gestures, directional buttons, and motion sensors (gyroscope), functioned properly and responsively on both the tested devices. In addition, the user interface (UI) displayed correctly without experiencing layout breaks across different screen sizes, both medium and large.

The scoring system operated in accordance with the game logic, whereby points continued to accumulate as long as the player survived the game. Sound effects, such as vehicle horns and collisions, as well as character animations, were also presented and synchronized with in-game events. From these findings, it can be concluded that *Highwash Rush* is ready for broader testing through a beta stage or limited APK release for end users.

4.3. Discussion

Based on the implementation and testing results, it can be concluded that the development process of *Highwash Rush* proceeded according to the initial design and methodological frameworks. All of the planned core features—such as the main menu interface, character controls, obstacle system, scoring mechanism, sound effects, and game-over and restart functions—were successfully implemented and operated effectively on the two Android devices tested.

In terms of functionality, the Black Box Testing results demonstrated that all major features performed as expected, with no critical bugs identified. Furthermore, Alpha Testing indicated that users were comfortable with the game controls and simplified interface. This suggests that the game successfully met most targeted functionality and user experience indicators.

Overall, the features designed during the initial stage were successfully implemented in a functional and stable manner, consistently demonstrating their reliability throughout testing. This confirms that the MDLC-based development approach is highly effective in producing high-quality educational and entertainment games on mobile platforms.

To further clarify the connection between the research problems and the implementation results, Table 5 presents a comparison.

Table 5. Comparison of Formulations and Results

Problem Statement (Chapter 1)	PU
How to design and implement a 2D mobile game with a vehicle obstacle avoidance theme?	The game was successfully designed and implemented through the MDLC stages and can be played smoothly on Android.
What features need to be developed to make the game interesting and challenging?	Dual control features (sensors, gestures, buttons), progressive scoring system, sound effects, and animations were successfully implemented.
How is the testing process to ensure that the game runs smoothly on mobile devices?	Testing was conducted using the Black Box and Alpha Testing methods; the results showed stable game performance.

Based on the results of implementation and testing, and as reflected in the comparison between the research problems and the development outcomes presented in Table 5, it can be concluded that all key aspects of this project have been successfully achieved. Each problem statement formulated at the beginning of the study was effectively addressed through a structured development process guided by the MDLC method. This process encompassed concept planning, interface and gameplay system design, media asset collection, and implementation, testing, and distribution.

Highwash Rush was designed and implemented with a focus on interactivity, ease of control, and a simple visual presentation. Features such as character control, obstacle spawning, progressive scoring, sound effects, and game-over and restart screens were successfully executed. The results of Black Box and Alpha Testing confirmed that the system operated stably and provided a satisfactory user experience at the initial development scale. Users reported that the game is easy to play and engaging, although further refinement was suggested, particularly in relation to the user interface (UI) and the diversity of sound effects.

Therefore, *Highwash Rush* not only fulfills the technical and functional objectives of this study but also demonstrates the potential for further development as both an entertainment and educational mobile-based medium. This discussion reinforces the conclusion that the MDLC method is effective for the development of interactive multimedia applications, particularly simple Android-based games targeted at adolescent users

5. CONCLUSION

Based on the outcomes of the design, implementation, and testing phases, it can be concluded that the development of *Highwash Rush* was successfully executed using the Multimedia Development Life Cycle (MDLC) method in a systematic manner. All stages of the MDLC, from concept to distribution, were carried out effectively, resulting in a functional 2D mobile game centered on vehicle obstacle avoidance, which operates optimally on Android devices. The core features, including character control, obstacle spawning, progressive scoring, sound effects, and menu and game-over screens, were successfully implemented. Black Box Testing confirmed that all features functioned as intended without significant bugs, while Alpha Testing demonstrated that users found the controls, simple interface, and clear gameplay flow satisfactory. Overall, the game achieved its primary development objectives, namely creating an interactive learning and entertainment medium that is lightweight, easy to play, and suitable for adolescent users. The findings also affirm that the MDLC approach is effective for the development of unity-based 2D games.

Although *Highwash Rush* has been successfully developed and tested, several areas warrant further enhancement: (1) Improvement of Visuals and User Interface (UI): The current graphics and interface elements are relatively basic. It is advisable to refine the design to achieve a more professional and visually appealing appearance; (2) Integration of New Features: The game could be expanded by incorporating features such as level systems, leaderboards, or reward mechanisms to enhance player motivation and provide greater challenges; (3) Development of Sound Effects and Music: Additional variations in sound effects and background music are recommended to enrich the gameplay atmosphere and enhance the overall user experience; (4) Broader Testing: Wider-scale testing with a larger and more diverse group of respondents is necessary to obtain more representative evaluations of the game's quality; and (5) Distribution and User Feedback: Following this initial development phase, the game should be released on a limited public scale to gather direct feedback from end users.

Based on the results of the design, implementation, and testing, it can be concluded that the development of *Highwash Rush* was successfully executed using the Multimedia Development Life Cycle (MDLC) method in a systematic manner. All MDLC stages, from concept to distribution, were implemented effectively, resulting in a 2D mobile obstacle-avoidance game that functions optimally on Android devices.

Considering the suggestions outlined above, the next version of *Highwash Rush* is expected to deliver added value in terms of technical performance, visual presentation, and the overall user experience..

Ethical approval

This study was conducted in accordance with the ethical principles outlined in the Declaration of Helsinki.

Informed consent statement

All participants were informed of the purpose of the study, and informed consent was obtained prior to the data collection. Participation was voluntary, and all responses were kept confidential and used solely for academic-research purposes.

Authors' contributions

CNA contributed to the conceptualization, research design, and overall supervision of this project. She also served as the corresponding author and coordinated communication during the submission and review processes. YY contributed to the design and implementation of Unity-based game mechanics, including obstacle avoidance features and control systems. IFKF contributed to material collection, testing, and documentation of the MDLC stages. DW contributed to the assembly of game assets, sound integration, and refinement of the final version of the manuscript.

Disclosure statement

The authors declare no potential conflicts of interest was reported by the author(s).

Data availability statement

The data presented in this study are available on request from the corresponding author due to privacy reasons.

Funding

This study received no external funding.

Notes on Contributors

Charmiyanti Nurkentjana Aju
https://orcid.org/0009-0002-3032-1094
Charmiyanti Nurkentjana Aju is a lecturer in the Multimedia Engineering Technology Department at Politeknik Digital Boash Indonesia. She has professional experience as an IT practitioner and has a strong background in academia and teaching. She has also participated in various research projects in the fields of Information Technology and Multimedia.

Yusran Yasir
Yusran Yasir is a Diploma IV student in Multimedia Engineering Technology at Politeknik Digital Boash Indonesia, with professional experience as a Graphic Designer in a modern Muslim fashion company. He is proficient in Adobe Photoshop, Illustrator, Figma, and Canva, with expertise in graphic design, UI/UX design, logo creation, and banner design. Currently, Yusran is actively participating in bootcamps and seeking internship or volunteer opportunities to further develop his skills in the creative design field.

Indah Fajrin Kamil Falaq
Indah Fajrin Kamil Falaq is a Diploma IV student in Multimedia Engineering Technology at Politeknik Digital Boash Indonesia. She has strong interests in graphic design, UI/UX, as well as website, application, and game development. She enjoys designing social media application interfaces and creating various types of posters, including promotional materials. One of her completed projects involved designing the user interface of an entertainment website for video and music content. Indah is a responsible individual with great enthusiasm for learning new things, particularly in the field of multimedia.

Dwi Wibisono
Dwi Wibisono is a Diploma 4 student in the Multimedia Engineering Technology Study Program at the Boash Indonesia Digital Polytechnic. His academic interests include graphic design, broadcasting, and web development. Dwi enjoys designing posters, banners, logos, and more. He is also an active student and eager to learn new things.

REFERENCES

Aleem, S., Capretz, L. F., & Ahmed, F. (2016). Game development software engineering process life cycle: A systematic review. *Journal of Software Engineering Research and Development, 4*(1), Article 6. https://doi.org/10.1186/s40411-016-0032-7

Aryani, D., Noviandi, N., Siti Fatonah, N., & Akbar, H. (2024). Implementation of the Multimedia Development Life Cycle (MDLC) in solar system application design. *International Journal of Science, Technology & Management, 5*(4), 726–732. https://doi.org/10.46729/ijstm.v5i4.1123

Atika, S. (2025). Recent Trends and Innovations in Elementary School Educational Game Development: A Literature Review. *Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP)*, *8*(1), 53–63. https://jurnal.unprimdn.ac.id/index.php/JUTIKOMP/article/view/6660

Binanto, I. (2010). *Multimedia digital: Dasar teori dan pengembangannya*. Andi. (ISBN 978-979-29-1328-6)

Ferdinand, K., Jonathan JM, K., & Haris, D. A. (2024). Innovative Approach of 2D Platformer Mobile Game Development "Super Journey." *Teknika*, *13*(2), 233–238. https://doi.org/10.34148/teknika.v13i2.857

Jitendra, M. S. N. V., Amiripalli, S. S., Surendra, T., Rao, R. V., & Chowdary, P. R. (2021). A study on game development using unity engine. *AIP Conference Proceedings*, *2375*(October). https://doi.org/10.1063/5.0066303

Maulana, D., Firmansyah, A., & Mawarni, S. (2022). Implementasi Metode Multimedia Development Life Cycle pada Game Visual Novel "Sebelum Kamu Membenciku." *Jurnal Teknologi Informatika Dan Komputer*, *8*(2), 337–351. https://doi.org/10.37012/jtik.v8i2.1283

Poondej, C., & Lerdpornkulrat, T. (2019). Gamification in E-learning: A moodle implementation and its effect on student engagement and performance. *Interactive Technology and Smart Education*, *17*(1), 56–66. https://doi.org/10.1108/ITSE-06-2019-0030

Prayudha, I. T., & Chotijah, U. (2024). Pengembangan Game Edukasi 2D Mata Pelajaran Ipa Menggunakan Unity Berbasis Mobile. *Jurnal Teknik Informatika Dan Komputer*, *3*(2), 46–52. https://doi.org/10.22236/jutikom.v3i2.15933

Putra, E., Nugraha, A. R., Rahmadi, F., & Habibi, F. (2023). Implementation of MDLC method in development of game learning media. *AIP Conference Proceedings, 2641*, 040003. https://doi.org/10.1063/5.0128325

Saputra, R. K., & Setiawan, K. (2024). Pengembangan Game 2D Platformer Berbasis Microbit Menggunakan Unity. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, *5*(1), 20–25. https://doi.org/10.35870/jimik.v5i1.420