International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5535~5542
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5535-5542

Platforma: a modular and agile framework for simplified platformer game development

Rickman Roedavan[1], Abdullah Pirus Leman[2], Bambang Pudjoatmodjo[1]
[1]Multimedia Engineering Technology Program, Faculty of Applied Science, Telkom University, Bandung, Indonesia
[2]Faculty of Information Sciences and Engineering, Management and Science University, University Drive, Shah Alam, Malaysia

Article Info

Article history:

Received Oct 31, 2024
Revised Jul 29, 2025
Accepted Sep 14, 2025

Keywords:

Agile framework
Game development
Game experience questionnaire
Game framework
Platformer

ABSTRACT

Research on game development frameworks has been extensively conducted; however, most frameworks are still too general. Conventional game frameworks are challenging for students who are new to game development, especially with their limited information and skills. Beginner game developers should ideally be guided by a practical and specific framework to help them better understand the structure of game development in a more directed manner. This paper proposes platformer modular and agile framework (Platforma) that specifically designed for platformer game development. The framework is built based on the atomic design model, breaking down each minor feature of a platformer game element and grouping these features into more specific modules. The framework was tested on three teams of students. Each team was tasked with developing a platformer game with a minimum of 15 levels of the reach game goals typology. Testing results involving 100 respondents using the game experience questionnaire (GEQ) indicated that the games developed had a positive aspect score of 3.48 and a negative aspect score of 2.65. Overall, these results suggest that the Platforma can serve as an effective guide for beginners in developing platformer games.

Corresponding Author:

Rickman Roedavan
Multimedia Engineering Technology Program, Faculty of Applied Science, Telkom University
Jl. Telekomunikasi No. 1, Terusan Buah Batu Bojongsoang, Bandung, West Java 40257, Indonesia
Email: rikman@telkomuniversity.ac.id

1. INTRODUCTION

A game is an interactive multimedia product that combines various disciplines, including visual, narrative, and programming elements. A game's development calls for both technical programming and a visual aspects [1]. Generally, games are developed through the game development life cycle (GDLC), which outlines the stages from initial concept to final product launch [2]. This cycle emphasizes planning, design, development, testing, and overall implementation of the game. However, in practical terms, developers often use a framework-based approach that aids in constructing the technical aspects of the game more specifically. Such frameworks provide structural guidance to help developers design, implement, and test games in alignment with the desired functional and aesthetic aspects [3].

The mechanics-dynamics-aesthetics (MDA) framework is one of the most influential and widely used frameworks in game development [4]. This framework offers an approach that enables developers to manage the mechanical, dynamic, and aesthetic components as three core elements supporting the gameplay experience. Although this framework has become a primary reference, various studies indicate some limitations, particularly in providing guidance to maximize player experience [5].

Several previous studies have expanded upon the MDA framework, one of which is design-play-experience (DPE) framework. This framework guides developers to consider how players interact within the game, not only in terms of mechanics but also in how these experiences impact player engagement [6]. Additionally, there is the expanded DPE, designed to address limitations in capturing the psychological and emotional variables present during gameplay. The expanded DPE enhances the experiential dimension by adding richer analytical components regarding players' emotional responses and the impact of interactive experiences on satisfaction and motivation levels [7]. Another framework is core elements of the gaming experience (CEGE). This framework focuses on the core elements that shape the gaming experience like challenge, fantasy, curiosity, and control [8]. The Table 1 summarizes the similarities and differences among the five main frameworks used in game development.

Table 1. Game framework comparison

Aspect	MDA	CEGE	DPE	Expanded DPE
Advantages	Simplifies gameplay elements	Comprehensive focus on player enjoyment	Emphasizes player interaction	Deepens player experience by considering psychological factors
	Provides a structured approach for mechanics, dynamics, and aesthetics	Improved player retention	Focuses on long-term engagement	Adds analysis of emotional responses and satisfaction
	Useful for evaluating gameplay effectiveness	User-centered design	Suitable for iterative design and testing	Enhances player motivation insights
Limitations	Limited in addressing player experience depth	Lack of specificity for game mechanics	Complex for beginners to implement	Requires specialized skills for effective implementation
	Lacks detailed guidance on player engagement	Limited focus on social interaction	Requires more steps, which may overwhelm novice developers	Increases development time due to added analytical layers
	Not fully comprehensive for advanced player interaction analysis	Challenges in measuring success of each element	Less practical for small-scale projects	Demands extensive testing and evaluation processes

In general, each framework has strengths in providing technical and structural perspectives, yet they are not entirely effective as guides for students who are new to game development. Since most frameworks are not specific, this results in a lack of practical and measurable guidance for beginners, especially in learning processes that require clear, evaluable outcomes [9]. Effective learning should include specific achievement indicators to enable comprehensive evaluation of the learning process [10].

This research aims to develop a framework called platformer modular and agile (Platforma), designed specifically to facilitate learning in platformer game development. The platformer genre was selected as it is considered a foundational genre in game development that is relevant and allows students to understand a more comprehensive game structure before progressing to other genres. This framework is expected to provide a specific modular structure, making it easier to measure learning effectiveness and outcomes.

The structure of this paper is as follows: the methodology section discusses the research methods used in developing the Platforma framework. The results and discussion section presents the results of testing three platformer games developed by three student teams. Finally, the conclusion section summarizes the research findings and offers suggestions for future research.

2. METHOD

2.1. Analysis

Development of the framework begin with adopting the atomic design model, a method that breaks down a product into its smallest elements, making each component easier to program [11]. This method enables developers to identify the core elements and mechanics of a game, especially in platformers, by analyzing each component at an atomic level. In this research, we conducted a comparative analysis of ten popular platformer games: Mario Bros, Sonic the Hedgehog, Donkey Kong, Rayman, Celeste, Shovel Knight, Hollow Knight, Metroid, Cuphead, and Mega Man. These games are considered foundational in the platformer genre and have been released across various gaming consoles [12]. From these games, we aimed to extract common mechanics and distinctive features that could be simplified and replicated to build a modular framework applicable to beginners in game development [13]. Table 2 provides detailed insights into the core features of each of these ten platformer games.

The comparative analysis reveals that the goals in platformers are generally categorized as either reaching the finish line or defeating a final boss, which aligns with the traditional progression style of platformer games. Another common characteristic observed was movement, typically side-scrolling from left to right, along with jumping mechanics that are often extended with additional movements such as dashing, double jumps, or wall-kicking depending on the game [14]. Games like Mario Bros and Hollow Knight showcase extensive combat actions, where the character's ability to attack enemies serves as a central mechanic, further diversifying the platformer experience [14]. This variety in movement and action types highlights the genre's potential for layered, increasingly complex interactions as players progress through different levels [15].

Regarding level design, several platformers feature checkpoints to simplify progress and include "obtainables" such as coins or power-ups, to aid the player's journey. The combination of obstacles and collectible items enhances the engagement level by creating both challenges and rewards [16]. Each game analyzed also incorporates distinctive styles of level design, such as dynamic or static backgrounds, adaptive obstacles, and varied enemy interactions [17]. Such features contribute to the uniqueness of each platformer and establish a set of elements to be included in a modular framework that beginner developers can use as a guideline for constructing their own platformer games [18].

Table 2. Game mechanic comparison

Name	Goal	Movement	Action	Obtainable	Obstacle	Special	Level
Mario Bros	Reach flag	Walk, run, crouch, jump, climb	Step on enemy, shot	Mushroom, coin, star	Enemy body, enemy shoot, moving platform, fire	Big when eat mushroom, invisible when touch star	Checkpoint, death, pipe teleport
Sonic	Reach finish	Walk, run, crouch, jump, climb, dash, spin, roll	Spin attack	Rings, shield	Enemy body, spikes, moving platform	Speed boost, invincibility when shield obtained	Checkpoint, loop, springs
Donkey Kong	Reach top	Climb, jump, walk	Barrel throw, hammer attack	Hammer	Barrel, fireball, spring jump	Temporary invincibility with hammer	Multiple floors, ladders
Rayman	Reach goal	Walk, run, jump, glide	Punch, kick	Power-ups, lumps	Enemies, moving platforms, spikes	Gliding ability with certain upgrades	Multi-zone checkpoints, ropes
Celeste	Reach summit	Run, dash, climb, jump	None	Strawberries	Spikes, falls, moving platforms	Dash recharge when landing, additional air	Rooms, secret areas
Shovel Knight	Defeat boss	Walk, jump, downward strike	Slash, downward strike	Gems, health potions	Spikes, enemies, fire	Invincibility with specific armor	Checkpoints, treasure rooms
Hollow Knight	Explore areas	Walk, run, jump, wall jump, climb	Slash, spell cast	Geo, health potions	Enemies, spikes, moving platforms	Power-ups that increase health and abilities	Checkpoints, hidden rooms
Metroid	Reach target	Walk, run, jump, morph ball	Shoot, bomb, missile	Energy tanks, missiles	Enemies, lava, doors	Morph ball for accessing small areas	Save stations, elevators
Cuphead	Defeat boss	Run, jump, dash	Shoot, parry	Coins, power-ups	Enemies, projectiles, moving platforms	Parry grants extra abilities	Boss rooms, secret areas
Mega Man	Defeat boss	Walk, run, jump, slide	Shoot, charge shot	Health, energy capsules	Enemy shots, spikes, moving platforms	Various powers based on defeated bosses	Checkpoints, boss doors

2.2. Proposed model

Based on the previous analysis, this research proposes the Platforma framework that divides platformer game development into four layers: core, primary mechanics, secondary mechanics, and logic. the core layer contains the essential elements required for any platformer game to establish a solid foundation: character, level, and goals. The character component within the core layer centers on basic movement and action mechanics essential to platformers [19]. The level component defines the layout and design of each stage, focusing on strategic placement of obstacles, platforms, and enemies [20]. The goals component draws from ten imperative game design typologies [21]. Each component of this layer is critical to the structure and functionality of a platformer, enabling developers to organize and build upon core gameplay mechanics with clarity. Platforma framework, illustrated in Figure 1.

The primary mechanics layer contains the character standard movement capabilities (such as walk, run, and jump) and basic actions (like slash, shoot, hit, and die). For level, fundamental requirements include a static or sky background, static obstacles (for example, platforms that characters can jump on), and obtainable items (such as coins to increase scores). The goals component, in the primary implementation of Platforma, only the reach typology is used; this means that the game is completed once the player reaches the endpoint, making it suitable for simplified game programming and design.

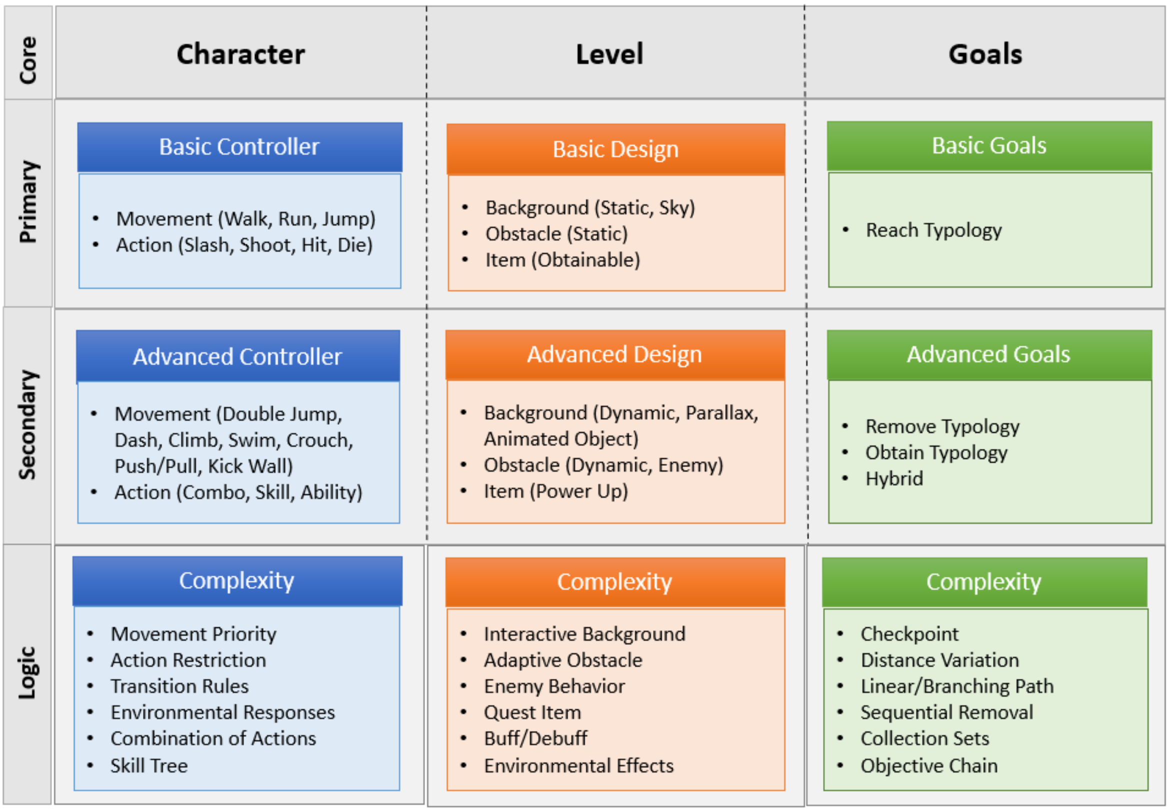

Figure 1. Platformer modular and agile framework (Platforma)

The secondary mechanics layer introduces advanced actions that typically require more sophisticated coding. For character, these secondary actions include more complex movements like double jump, dash, climb, swim, crouch, push/pull, and wall kick. Additional actions such as combo moves, skills, and special abilities enhance gameplay complexity and engagement. For level, secondary mechanics include advanced elements like dynamic backgrounds, parallax scrolling, animated objects, and obstacles that interact with the player, such as moving platforms or enemies. The goals component in this layer adds remove typology, obtain typology, and hybrid typologies, extending gameplay by allowing players to interact with the environment in more varied and challenging ways.

The logic layer is critical for guiding developers on implementing mechanics complexity within the game structure. For basic movement mechanics, logic considerations such as movement priority, determining which action (walk or run) takes precedence based on speed parameters. For action restrictions, logic rules dictate when a character can attack or move and whether certain actions can be performed simultaneously. Additionally, transition rules address visual continuity, ensuring smooth transitions between animations (for example, handling cases where a character transitions from idle to jump without any visual stutter or disruption).

Within level design, logic for effects like parallax scrolling is crucial to create visually appealing backgrounds. Another important aspect is adaptive obstacle logic, which defines behaviors such as moving or attacking obstacles that dynamically adjust to the player's actions or position. Logic for obtainable items dictates each item's impact on the player, such as score or health increases, along with how each item influences player movement or status effects.

Logic for goals is essential to define and track game completion. At a basic level, checkpoints ensure players progress by saving their position, and distance checks validate whether a player has reached specific locations. For more advanced gameplay, logic can support multi-objective goals, such as completing a set collection of items or activating objects in a specific sequence. Complex goal structures, including branching paths or both linear and non-linear goals, further enhance the player experience by offering a richer, more varied progression system [22]. These logic rules empower beginner developers to implement and expand their games' goal structures, creating a more engaging and immersive experience for players.

3. RESULTS AND DISCUSSION

To evaluate the Platforma framework, three teams of third-year university students with a foundational knowledge in game programming were formed. Each team comprised three members: a programmer, an artist, and a narrative designer. The teams were tasked with developing a 15-level platformer game over a span of three months using the unity game engine. Their primary objective was to implement primary platformer mechanics, including character movement, jumping, interaction with obstacles, collection of obtainable items, and achieving a "reach" goal to complete levels. To simplify development, the teams were permitted to utilize free or licensed assets from the unity asset store, allowing them to focus on game mechanics and level design.

Team A created a medieval-themed game featuring a knight on a quest to rescue a kidnapped princess. The storyline required players to navigate dungeons, overcoming various platform challenges and combatting enemies along the way. The primary mechanics focused on combat, exploration, and platforming obstacles. The game aimed to immerse players in a richly detailed medieval world, filled with atmospheric visuals and engaging soundscapes to enhance the sense of adventure. Figure 2 shows an example of gameplay developed by Team A, showcasing the knight's journey through perilous environments and intense enemy encounters.

Team B developed a game centered around a ghost character trapped in a haunted room. The ghost could possess objects to aid in escaping the strange environment, with the primary game mechanic focusing on the ghost's unique ability to manipulate items. As players navigate the eerie setting, they must solve puzzles and overcome supernatural obstacles to find a way out. Figure 3 presents a screenshot of the gameplay developed by Team B, highlighting the ghost's interaction with its surroundings and the hauntingly atmospheric design.

Lastly, Team C designed a slime-themed game where the player-controlled slime must reach a goal by interacting with environmental objects that alter its movement and trajectory. The core gameplay involved innovative interactions between the slime character and various objects, such as mushrooms that boost jumps and electricity objects that change speed or direction, creating dynamic puzzles for the player to solve. This unique mechanic encouraged players to experiment with different object interactions to discover the best path forward. Figure 4 showcases these interactions within a gameplay.

The results indicate that using the Platforma framework effectively streamlined the development process for platformer games. Although each team was tasked with creating 15 levels, the framework allowed them to concentrate on developing a single short level with core mechanics that could then be easily adapted and modified for subsequent levels. This modular structure significantly accelerated the development of additional levels and gameplay experience once the basic framework was established [23].

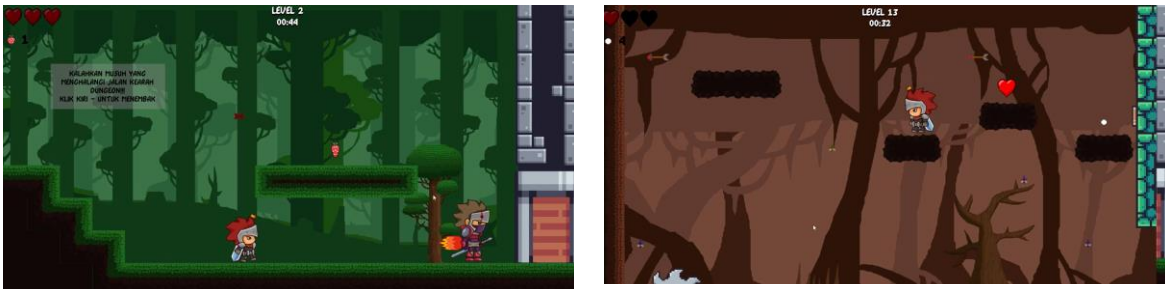

Figure 2. Game Knight Redemption created by Team A

Figure 3. Game Ghost Escape created by Team B

Figure 4. Game Slimetric created by Team C

The three games were tested with 100 university student respondents at Telkom University, comprising 70 males and 30 females. Each team promoted their game over a two-to-three-week period during a campus expo held in the university lobby, allowing respondents to playtest the games. Afterward, the respondents completed a Google Form survey based on the game experience questionnaire (GEQ). For this evaluation, only the core module of the GEQ, consisting of 33 questions, was used [24].

The questions were grouped into six categories, there are: Competence to measures the player's perception of skill improvement and the ability to overcome challenges; Sensory and imaginative immersion to evaluates the player's feeling of being absorbed in the game world; Flow to captures the sense of being fully engaged and losing track of time while playing; Tension/Annoyance to assesses the level of frustration or tension experienced; Challenge to gauges the perceived difficulty and stimulating nature of the game; and Positive affect/negative affect to reflects the player's positive or negative emotional response during gameplay. Table 3 shows overview of survey results.

Table 3. GEQ overview of the results for each game

Game	Competence	Sensory and imaginative immersion	Flow	Tension/annoyance	Challenge	Negative affect	Positive affect
Team A	4.05	4.03	3.18	2.60	2.87	2.23	4.23
Team B	3.11	3.03	2.85	2.73	2.76	2.84	3.17
Team C	2.95	3.17	2.99	2.67	2.94	2.86	3.03
Average	3.38	3.41	3.01	2.67	2.86	2.65	3.48

On average, the games scored 3.48 on positive aspects and 2.65 on negative aspects. In GEQ evaluations, a score above 2.5 for positive aspects suggests a favorable experience, while a score below 2.5 for negative aspects implies a less desirable experience. While all three games had average tension scores of 2.67, this suggests a somewhat high level of frustration among players, aligning with the challenge score average of 2.86. Despite the challenging nature, the average competence score of 3.38 indicates that most players were able to complete the games successfully, even with difficult and somewhat frustrating gameplay elements. These scores suggest that the games, while challenging, were ultimately enjoyable and engaging for most players, aligning with the intended design goals of the framework. From a game development perspective, these findings confirm that the Platforma framework effectively guided beginner developers in creating platformer games that balance challenge with player engagement. The feedback collected through the GEQ demonstrates that the framework's modular design and core mechanics provide a solid foundation for students, allowing them to produce functional, enjoyable platformer games despite limited prior experience. This makes Platforma a valuable educational tool for guiding beginner developers in creating structured, interactive platformer games.

4. CONCLUSION

Platforma framework has proven effective in guiding beginner developers in the creation of platformer games by breaking down core game mechanics into manageable, modular components. Through the development and testing of three unique games, the framework demonstrated its utility in providing structure, enhancing the learning process, and simplifying the adaptation of core mechanics across multiple levels. Positive feedback from players suggests that the framework supports engaging and immersive gameplay experiences, even when developed by novice teams. Moving forward, future work could explore expanding the framework to support other game genres, adding more advanced logic modules for complex game mechanics, and developing an AI-driven assessment tool to provide real-time feedback on game design quality. Further research may also investigate the potential of this framework in fostering collaboration between interdisciplinary teams, enhancing its effectiveness as a learning tool across various educational settings.

REFERENCES

[1] R. Roedavan, B. Pudjoatmodjo, Y. Siradj, S. Salam, and B. Q. D. Hardianti, "Serious game development model based on the game-based learning foundation," *Journal of ICT Research and Applications*, vol. 15, no. 3, pp. 291–305, 2021, doi: 10.5614/ITBJ.ICT.RES.APPL.2021.15.3.6.

[2] R. Hidayat, M. W. A. Bawono, and M. J. Alibasa, "Developing a horror game with game development life cycle method using unity game engine," *2023 International Conference on Data Science and Its Applications, ICoDSA 2023*, pp. 471–476, 2023, doi: 10.1109/ICoDSA58501.2023.10277028.

[3] Z. Guo and C. H. Lo, "An empirical framework for understanding a player's sense of agency in games," *International Journal of Human-Computer Interaction*, vol. 40, no. 19, pp. 5717–5736, 2024, doi: 10.1080/10447318.2023.2241286.

[4] B. Pudjoatmodjo, S. Salam, N. C. Pee, R. A. Rudavan, A. S. Prihatmanto, and A. Alomoush, "The 3d dyscalculia assessment game framework for dyscalculia identification," *International Journal of Computing and Digital Systems*, vol. 11, no. 1, pp. 451–461, 2022, doi: 10.12785/ijcds/110137.

[5] Z. Tang and B. Kirman, "Exploring curiosity in games: A framework and questionnaire study of player perspectives," *International Journal of Human-Computer Interaction*, vol. 41, no. 4, pp. 2475–2490, 2025, doi: 10.1080/10447318.2024.2325171.

[6] G. Karnchanapayap, "Activities-based virtual reality experience for better audience engagement," *Computers in Human Behavior*, vol. 146, p. 107796, 2023, doi: 10.1016/j.chb.2023.107796.

[7] L. Sun, M. Kangas, and H. Ruokamo, "Game-based features in intelligent game-based learning environments: a systematic literature review," *Interactive Learning Environments*, vol. 32, no. 7, pp. 3431–3447, 2024, doi: 10.1080/10494820.2023.2179638.

[8] S. Ghulamani, A. Shah, and K. Khowaja, "Review of components from frameworks and models for DGBL for history based courses," *University of Sindh Journal of Information and Communication Technology*, vol. 4, no. 1, pp. 9–16, 2020.

[9] I. A. Akour, R. S. Al-Maroof, R. Alfaisal, and S. A. Salloum, "A conceptual framework for determining metaverse adoption in higher institutions of gulf area: An empirical study using hybrid SEM-ANN approach," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100052, 2022, doi: 10.1016/j.caeai.2022.100052.

[10] S. Kirk and J. King, "EAP teacher observation: Developing criteria and identifying the forms of pedagogic practice they afford," *Journal of English for Academic Purposes*, vol. 59, pp. 1–12, 2022, doi: 10.1016/j.jeap.2022.101139.

[11] R. Roedavan, A. Pratondo, B. Pudjoatmodjo, and Y. Siradj, "Adaptation atomic design method for rapid game development model," *International Journal of Applied Information Technology (IJAIT)*, vol. 4, no. 2, pp. 93–102, 2020.

[12] K. Ferdinand, K. Jonathan JM, and D. A. Haris, "Innovative approach of 2D platformer mobile game development 'super journey,'" in *Teknika*, 2024, vol. 13, no. 2, pp. 233–238, doi: 10.34148/teknika.v13i2.857.

[13] S. Tomkinson, "Video games through the refrain: Innovation and familiarity," *Journal of Gaming & Virtual Worlds*, vol. 12, no. 3, pp. 287–302, 2020, doi: 10.1386/jgvw_00020_1.

[14] A. A. Gadir Ahmadov, Araz Aliyev, Gadir Ahmadov, "Integrating visual effects into 2d platformer games to enhance play experiencess," in *PAHTEI-Procedings of Azerbaijan High Technical Educational Institutions*, 2024, vol. 40, no. 05, pp. 291–300, doi: 10.36962/pahtei40052024-291.

[15] K. Kostelić, "Dynamic awareness and strategic adaptation in cybersecurity: A game-theory approach," *Games*, vol. 15, no. 2, p. 28, 2024, doi: 10.3390/g15020013.

[16] R. Dewan, R. Polishetty, N. Jagadam, M. K. Goyal, K. K. Ravulakollu, and B. Sharan, "Significance of state-of-art search engine in game development," *Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development, INDIACom 2023*, pp. 1134–1139, 2023, doi: 10.1109/TMM.2023.10112272.

[17] M. Lebron, R. G. Swab, and R. Bruns, "Students as game designers and developers: developing cooperative strategy board games to teach team leadership skills," *Organization Management Journal*, vol. 21, no. 1, pp. 41–48, 2024, doi: 10.1108/OMJ-03-2023-1783.

[18] M. Pichlmair and M. Johansen, "Designing game feel: A survey," *IEEE Transactions on Games*, vol. 14, no. 2, pp. 138–152, 2022, doi: 10.1109/TG.2021.3072241.

[19] S. Patil and S. Sorte, "2D game development," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 241–246, 2025, doi: 10.48175/ijarsct-25233.

[20] A. Geriş, *et al.*, "Revolutionising game design: The role of AI in enhancing player experience, interaction, and engagement," *Öneri Dergisi*, vol. 20, pp. 89-113, 2025, doi: 10.14783/maruoneri.1499524.

[21] R. Roedavan, Y. Siradj, and S. Stefany, "Educational game scenario model based on imperative game goal typology," *Journal of Games, Game Art, and Gamification*, vol. 8, no. 1, pp. 18–23, 2023, doi: 10.21512/jggag.v8i1.9497.

[22] J. Karapakdee and P. Wannapiroon, "Immersive digital storytelling learning experience with a metaverse gamification game platform to enhance game developer competency," *International Journal of Information and Education Technology*, vol. 13, no. 6, pp. 890–898, 2023, doi: 10.18178/ijiet.2023.13.6.1884.

[23] E. Handriyantini and S. Salem, "Designing user experiences in casual games to enhance product knowledge," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 3, pp. 488–495, 2025, doi: 10.33480/jitk.v10i3.5761.

[24] D. K. Madana, E. M. A. Jonemaro, and H. M. Az-Zahra, "Evaluation of the Raft Game user experience using the game method experience questionnaire (GEQ)," (in Bahasa), *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 10, pp. 4494–4503, 2021.

BIOGRAPHIES OF AUTHORS

Rickman Roedavan is the author of the best-selling "Unity tutorial game engine series." He is currently the head of multimedia engineering technology research labs (Metalabs) and a lecturer in application programming for games at Telkom University, Indonesia. He also serves as an instructor for international classes in collaboration with King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, Universiti Kuala Lumpur (UniKL), and Management and Science University (MSU), Malaysia. Rickman is actively involved in international research partnerships with institutions such as the Royal Melbourne Institute of Technology (RMIT), Australia, Multimedia University (MMU), and Universiti Teknologi Melaka (UTEM), Malaysia. Prior to his academic career, he was a senior programmer at Indonesian Aerospace (IAe), where he specialized in wargaming system and flight simulator development. He can be contacted via email at rikman@telkomuniversity.ac.id.

Abdullah Pirus Leman is an academician with more than 27 years of industrial experiences. Currently, he is working as research group leader at Management and Science University, Shah Alam, Malaysia. Abdullah Pirus is holding certified professional Autodesk® AutoCAD in 2013. He also expecting to complete Sijil Kemahiran Malaysia, Sijil professional game development (from UTM SPACE & Todak Academy) in August 2023. His highest achievement was he gained the intellectual property rights (IPR) for his virtual reality (VR) application which focused on teaching tool for the subject design for secondary school student. He also supervised with a number of undergraduate final year projects either at diploma or bachelor level. Interest areas: game design, 3D animation, 3D modelling, storytelling, virtual reality, architectural visualization. Expertise areas: 3D game asset development, 3D animation, 3D modelling, storytelling, architectural visualization, computer aided design (CAD). He can be contacted at email: abdullah_pirus@msu.edu.my.

Bambang Pudjoatmodjo earned his Ph.D. in 2024 from Universiti Teknikal Malaysia Melaka (UTeM), specializing in interactive media and game design, following a master's degree in software reliability from Langlangbuana University. Since starting his career as a lecturer in 2002, he has taught various subjects in computer science and multimedia development. Over time, his passion for storytelling led him to explore film scenario writing, which, in 2011, inspired a deeper interest in scenario design specifically for game level development. His research has since expanded into interactive and immersive technologies, including virtual reality (VR) and augmented reality (AR), with a focus on enhancing user experience through narrative and environment design. He continues to actively contribute to the field of multimedia education and is involved in several projects that integrate scenario-based learning within game development frameworks. He can be contacted at email: bpudjoatmodjo@telkomuniversity.ac.id.