JAIIT (Journal of Advances in Information and Industrial Technology)
Vol. 7, No. 2 Nov 2025, ISSN 2716-1935, E-ISSN 2716-1927

Design and Construction of the Action Role-Playing Game: Supply Odyssey Using the Game Development Life Cycle Method

Enggaling Aji Dewanata[*1)], Tito Bisma May Willis[2)]
[1)]Department of Information System, Universitas Mulia, Samarinda, Indonesia
[2)] Department of Logistics Engineering, Institut Teknologi Kalimantan, Balikpapan, Indonesia
[1*]enggaling@gmail.com, [2]tito.willis@lecturer.itk.ac.id

ARTICLE INFO

Article history:
Received 06 October 2025
Revised 6 November 2025
Accepted 20 November 2025
Available online 26 November 2025

Keywords:
Action Role-Playing Game
Game Development Life Cycle
Line of Sight
Supply Odyssey

ABSTRACT

The rapid development of the global gaming industry has made games not only serve as entertainment but also as a medium for learning, social interaction, and game-based entrepreneurship. One of the most widely favoured genres is the Action Role-Playing Game (Action RPG), which combines action elements with RPG mechanics. This research aims to design and develop a 2D video game titled "Supply Odyssey" in the Action RPG genre using the Godot Engine as the primary development platform. Godot was selected due to its open-source nature, free licensing, and strong support for both indie and professional developers. The research employs the Game Development Life Cycle (GDLC) methodology, incorporating the Line of Sight (LOS) technique to model enemy (NPC) behaviour. The implementation of LOS is expected to increase combat realism and provide greater tactical depth in real-time encounters by enabling players to strategize according to enemy visibility ranges, select optimal positions, and engage in more dynamic interactions. Supporting data were collected through observations on game platforms and developer and gamer community forums to identify relevant trends, preferred game concepts, and engaging gameplay patterns. The results of this research are expected to produce an Action RPG game that not only offers entertainment but also contributes to the development of Godot-based games and serves as a reference for applying GDLC and LOS methods in similar game development projects.

1. Introduction

The global development of games in recent years has been quite rapid, so that games are now not only for entertainment but also can be used as learning tools, social interaction, and business. Many companies sell their games to be marketed in various countries, and one of the most popular game genres is the Action RPG (Role-Playing Game) [1]. Action RPG games are a video game genre that combines action gameplay with war elements from RPG games [2]. In Action RPGs, players typically control the main character directly in battle, while gaining experience, improving character abilities, and managing inventory as is typical in RPG games.

Based on observations and surveys carried out on several major gaming-related websites, the current level of interest in Action RPG games is significantly high. Usually, those interested in this genre use PC devices [3]. To play games, players usually need a gaming platform application such as Steam, Epic Games, etc. These platforms function to provide a variety of games in various genres, in addition to making it easier for game makers to publish, market, and monetize [4].

One of the challenges in game development is choosing the right platform and tools to realize a developer's creative vision. In recent years, Godot Engine has emerged as a highly promising game development platform [5]. Godot Engine is an open-source game engine with a wide range of features and ease of use, making it a top choice for many indie and professional developers [6].

Although each game engine has its own strengths and weaknesses, Godot was selected because its open-source nature, lightweight architecture, and efficient 2D development workflow offer advantages

over other engines, making it highly suitable for the needs of this research. In some games, Godot Engine is still rarely used compared to Unity Engine and Unreal Engine. The researchers recommend broader adoption of Godot Engine due to its open-source accessibility and royalty-free licensing. If the developed videogame is to be sold, then the best choice is Godot Engine, as all profits are not shared with the owner [7].

Table 1. Comparative of Godot Engine, Unity Engine, and Unreal Engine

Criteria	Godot Engine	Unity Engine	Unreal Engine
Programming Languages	Uses GDScript (Python-like), C#, and C++. Simple and lightweight scripting makes development accessible [5].	Uses C# with Mono; widely known for rapid prototyping and mature API [5].	Uses C++ and Blueprint visual scripting; enables both high-performance and designer-friendly workflows [8].
License / Cost Model	MIT open-source license; free to use, modify, and distribute with no royalties.	Commercial license; free tier plus paid tiers depending on revenue and team size.	Free up to a revenue threshold, then royalties apply depending on commercial use.
2D Game Performance	Highly efficient for 2D due to low overhead and dedicated lightweight 2D pipeline [5].	Good performance, though its general-purpose engine introduces additional overhead compared to Godot [5].	Supports 2D but often considered overpowered for simple 2D projects; heavier runtime footprint [8].
3D Graphics & Rendering	Capable for mid-range 3D but not yet comparable to AAA engines in photorealistic output [5].	Flexible pipelines (Built-in, URP, HDRP) enabling scalable rendering from mobile to high-fidelity PC/console [8].	Best-in-class photorealism; advanced features such as Lumen, Nanite, and cinematic tools dominate AAA production [8].
Learning Curve / Productivity	Easiest learning curve; GDScript and scene-tree architecture allow rapid iteration and beginner accessibility [5].	Large documentation base, C#, Asset Store, and strong toolchain enable high productivity [8].	Steeper learning curve due to C++ complexity; Blueprint helps non-programmers build logic faster [8].
Community & Ecosystem	Strong open-source community; engine is fully modifiable, suitable for research and academic use [5].	Industry-leading ecosystem: Asset Store, plugins, third-party tools, extensive tutorials [8].	Marketplace supports high-end development; widely used in AAA, film, architecture, and simulation projects [8].
Limitations	Smaller asset ecosystem; 3D toolset still maturing compared to Unity/Unreal [5].	Licensing costs can scale; engine optimizations required for very large projects [8].	High complexity, larger runtime, C++ difficulty, and royalty structure [8].

In addition, recent studies in the field of game development emphasize the importance of combining engine capability with optimized gameplay design to enhance player experience and engagement [9]. demonstrated that effective integration between scripting languages and engine frameworks improves development efficiency in 2D RPG projects. Similarly, [10] highlighted the potential of adaptive artificial intelligence to dynamically adjust game difficulty, thereby maintaining player engagement over time. These approaches align with the current research objective of implementing an efficient development framework through the Godot Engine while enhancing interactive gameplay through the Line of Sight mechanism.

Based on the explanation above, the researcher wants to develop a video game with the Action RPG genre. The idea that will be implemented in the video game is the adventure of a character in a world whose task is to complete a Quest given by an NPC (Non-Player Character). The video game

 DOI: https://doi.org/10.52435/jaiit.v7i2.720

will be playable on a computer using the Windows operating system, and using a Keyboard and Mouse to control the character to be played. The developer will use the Godot Engine to create the video game. The programming language that will be used is GDscript, and Visual Programming. The video game will be 2-dimensional. Based on this background, the author wants to design, build and record the process through this research with the title "Design and Construction of the Action RPG Game "Supply Odyssey" Using the Godot Engine".

2. Methods

A video game is a form of electronic game played on an electronic device such as a computer, game console, or smartphone. Typically, video games involve player interaction with visual displays displayed on a screen, and often involve specific goals or missions that the player must achieve [11]. Game genre is a classification used to group video games based on similarities in gameplay style, themes, or certain gameplay elements. The Action RPG (Role-Playing Game) genre is a type of video game that combines role-playing game (RPG) elements with real-time action [12]. In Action RPGs, players control the main character and engage in direct combat within an active and dynamic game environment. Characters typically possess special abilities or skills that can be used in combat to defeat enemies or complete missions [13]. In addition, players often have the freedom to develop their characters by upgrading their stats, skills, or collecting items. Battles in Action RPGs typically take place in real time, unlike traditional RPGs that use turn-based combat [9]. This provides a sense of action and speed in gameplay, while maintaining RPG elements such as character progression and a deep narrative. Examples of action RPG games include "The Legend of Zelda: Breath of the Wild," "Dark Souls," and "The Witcher 3: Wild Hunt" [14]. Godot Engine is a free, open-source application that provides tools for video game development. It supports a scene and node system. Every object created in a video game is represented as a node. Each node can be reused and placed in other scenes.

2.1. Technical Implementation of Line of Sight

The Line of Sight (LOS) system was implemented using Godot's Area2D and RayCast2D nodes. Each enemy NPC is equipped with a forward-facing RayCast2D that detects the player character when within a specified range and unobstructed by collision layers. Detection triggers a state transition from idle to pursuit mode. The field of view is limited to a 90-degree cone to simulate realistic vision. This approach ensures low computational overhead while maintaining strategic depth in gameplay.

This research method uses the GDLC (Game Development Life Cycle) method and applies the LOS (Line of Sight) method to NPC enemy behavior. Applying the LOS (Line of Sight) method to this Action RPG allows players to frequently engage in real-time combat with enemies [9]. By taking line of sight (LOS) into account implemented through a raycasting based visibility check that determines whether obstacles block an enemy's field of view players are able to develop more sophisticated and tactically informed strategies during combat, as positional decisions and movement patterns directly influence detection and engagement outcomes. Players can also choose optimal positions to attack enemies or evade attacks more effectively, which can influence the outcome of battles [12]. To obtain data and literature studies, researchers conducted observations on several game platforms, forums and gamer communities such as gamedev.net, r/gamedev, discord servers, etc. So that researchers can find out the game genres, game concepts, and game flow that are currently in high demand by gamers. This selection was supported by comparative studies showing Godot's faster prototyping and lower runtime overhead. [11] found that Godot offers faster prototyping and lighter runtime performance compared to Unity for indie-scale 2D games. [12] proposed a hybrid Line of Sight and A* pathfinding model that improves NPC tracking accuracy and reduces computational load in real-time gameplay. Moreover, [13] confirmed that visual design and character feedback significantly influence player immersion in Action RPG settings. These findings reinforce the methodological decision to employ GDLC and LoS approaches for both development efficiency and user experience enhancement.

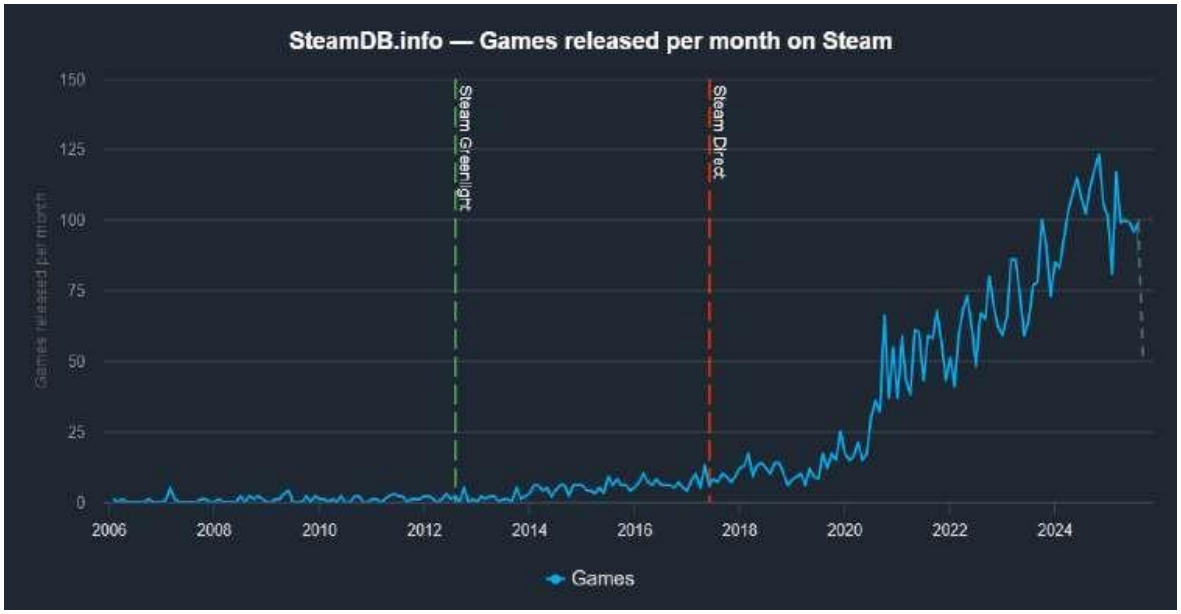


Figure 1. SteamDB Data Action RPG Games [15]

Data from SteamDB shows (Figure 1.) that the number of Action RPG games released per month continues to increase, indicating growing player interest and demand for this genre. The increasing number of releases demonstrates that developers increasingly recognize the significant market opportunity, leading many studios to strive to create new games to meet the growing demand of players. This development also demonstrates the growing popularity of Action RPGs among the gaming community.

3. Results and Discussions

3.1. Initiation

Table 2. Initiation

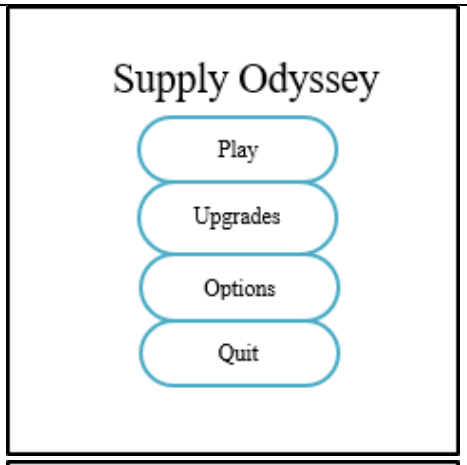
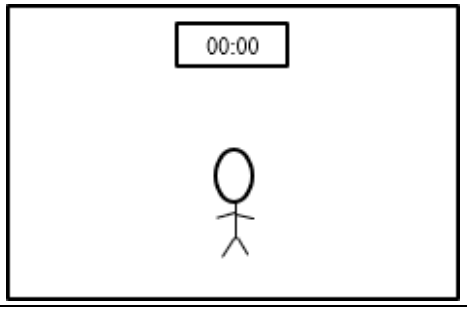
Rubric	Explanation
Game Concept	"Supply Odyssey" is an action RPG that combines adventure elements with strategic challenges and world exploration. The storyline follows the journey of a young courier navigating a fantasy world filled with mystery and danger, where they must uncover the cause of disrupted supply routes while facing hostile creatures and an emerging antagonist threatening the stability of the realm. Players will lead their character through various adventures, encountering NPC characters, uncovering dungeon secrets, and battling powerful enemies.
Game Mechanism	• Tactical Combat: Real-time combat that requires strategy, reflexes, and the use of character-specific skills to defeat a diverse range of enemies. • Open World Map Exploration: A vast, open game world to explore, with various points of interest, quests, and secrets awaiting players. • NPC Interaction: Players can interact with various NPCs in the game world, accepting missions, gaining information, or building relationships that influence the storyline. • Character Customization: The ability to upgrade and customize characters, including equipment, weapons, and skills, to suit individual playstyles and strategies.
Game Objective	• Completing Main Quests: Players must complete a series of main missions that follow the main storyline and uncover mysteries along the way. • Character Enhancement: Players must increase their character's strength and skills by completing missions, defeating enemies, and finding rare items. • World Exploration: Players are expected to explore the vast game world, discover hidden locations, and uncover hidden secrets. • Boss Battles: Players will face epic battles with powerful bosses, requiring precise skill and strategy to defeat them.

 DOI: https://doi.org/10.52435/jaiit.v7i2.720

3.2. Pre-Production

The pre-production phase determines gameplay, mechanics, storyline, character design, map design, and any assets needed. A storyboard is the game's storyline. This storyboard is essential for a game to have a clear purpose. The result of the storyboard is a story. In a world devastated by a witch's curse, monsters and golems appear endlessly, devouring every life they encounter. You are Mystro, the last adventurer of the Order of the Guardians of Light, tasked with finding the source of the darkness and destroying it. Your journey begins in a desolate place, accompanied only by the few tools you carry. Over time, the hordes of creatures grow stronger, urging you to survive and thrive. With each battle, your power increases, unlocking legendary skills and weapons. However, time is running out. The curse is growing stronger. Can you complete Supply Odyssey before the world falls into darkness?

Table 2. Story Board Main Menu & In-Game

Explanation	Sketch
The Main Menu page is the initial screen of the game. Players will be directed to the main menu, which contains the Play button to start the game, Upgrades to increase passive skills, Options to adjust game configuration, and Quit to exit the game.	
This page displays the in-game screen, where players will battle monsters. A timer will run above to track how long the player has survived.	

In this asset needs analysis stage after making the storyboard (Table 2.), all the assets needed to make this game are character assets, monsters, tilemaps, objects, and music audio.

3.3. Production

At this production stage, the game programmer will create coding for the gameplay mechanics that will be carried out when the required assets are fulfilled. The illustration (Figure 2.) shows a collection of visual and audio assets used during the development of Supply Odyssey. These assets include player and enemy sprites, environmental tiles, interactive objects, background layers, and various supporting effects such as particles and ambient sounds. Each asset was designed in a consistent pixel-art style to maintain visual coherence and aesthetic balance across all scenes. The selection of lightweight 2D assets was made deliberately to optimize performance, improve rendering efficiency, and reduce memory load, ensuring that the game runs smoothly even on lower-end systems. Every asset went through multiple stages of testing and refinement within the Godot Engine to achieve visual stability, animation smoothness, and cohesive integration with the overall game design.

Figure 2. Examples of Asset Games

Figure 3. Main Menu Display

The main menu (Figure 3.) serves as the player's first interaction point with Supply Odyssey. It provides four essential navigation buttons Play, Upgrades, Options, and Quit that grant direct access to the game's primary features. The layout is designed with simplicity and clarity in mind, combining clean typography, well-structured alignment, and intuitive button placement to enhance user experience. Subtle background animations and ambient sound effects establish an engaging atmosphere while maintaining a lightweight performance footprint. The interface ensures that players can easily start a new session, modify settings, or exit the game without unnecessary complexity, emphasizing usability and accessibility as core design principles.

 DOI: https://doi.org/10.52435/jaiit.v7i2.720

Figure 4. Upgrades Menu Display

The upgrades menu (Figure 4.) enables players to enhance their passive skills and improve character performance through experience points collected during gameplay. This feature introduces a layer of strategy and player agency, allowing for personalized playstyles and diverse tactical approaches. Each skill upgrade directly affects gameplay parameters such as damage output, movement speed, and defense resilience. The menu's interface provides real-time feedback on available experience points, upgrade costs, and current skill levels, promoting strategic decision-making and long-term progression. By integrating progression and challenge, this system encourages replayability and sustained player engagement throughout the game.

Figure 5. Options Menu Display

The options menu (Figure 5.) provides players with a comprehensive set of configuration controls that allow customization of their gameplay experience. Available settings include SFX Volume, Music Volume, and display modes such as Fullscreen and Windowed. These adjustable parameters ensure that players can fine tune audio and visual performance to suit their hardware capabilities and personal

preferences. Beyond convenience, the configuration system also contributes to optimization helping maintain resource efficiency during gameplay. The focus on user adaptability demonstrates a commitment to accessibility and inclusivity, ensuring that Supply Odyssey remains enjoyable across a wide range of devices and performance levels.

Figure 6. Gameplay Display

The gameplay section (Figure 6.) represents the core experience of Supply Odyssey, where players engage directly with the game's combat and survival mechanics. During gameplay, the player controls the main character within a dynamic 2D environment filled with continuously spawning enemies. The objective is to survive for as long as possible while collecting experience points dropped by defeated monsters. These experience points are later used to upgrade skills and enhance the character's abilities. The interface includes a timer at the top of the screen, which tracks the player's survival duration, as well as indicators for health, experience, and available upgrades.

Figure 7. Skill Selection Display

The combat system operates in real time and is designed to emphasize reflex, positioning, and

 DOI: https://doi.org/10.52435/jaiit.v7i2.720

spatial awareness. Enemy AI behavior, powered by the *Line of Sight* (LoS) system, allows enemies to detect the player only when within their visible range, adding an element of stealth and tactical movement to each encounter. As time progresses, the difficulty gradually increases—enemies spawn more frequently, possess higher health, and inflict greater damage—requiring the player to continuously adapt their strategy. This gameplay design aims to create a balanced combination of action intensity and strategic depth. The 2D pixel-art environment enhances readability and maintains performance efficiency, while the Godot Engine ensures smooth execution of animations, collisions, and physics-based interactions. Overall, the gameplay structure provides a compelling and progressively challenging experience that embodies the essence of the Action RPG survivor-like genre.

The skill selection menu (Figure 7.) functions as a key progression mechanic in Supply Odyssey, appearing whenever the player gains enough experience points to level up. At each level increment, the system randomly generates two new skill options, and the player must choose one to acquire. This randomized selection process introduces an element of unpredictability and encourages adaptive decision making, as players must evaluate which ability best suits their current strategy and combat situation. Each chosen skill directly influences gameplay dynamics by modifying attack patterns, cooldown rates, defensive attributes, or mobility, thereby promoting diversity in tactical approaches and replayability. The interface is designed to present clear visual feedback, including skill icons, brief descriptions, and selection highlights to ensure that choices are made efficiently during fast paced gameplay. Moreover, the design balances simplicity and visual appeal through consistent pixel-art styling and color cues that correspond to different skill categories. This system not only enhances player engagement but also reinforces the core Action RPG identity of the game by rewarding progression, experimentation, and strategic growth over time.

3.4. Testing

The current testing phase focuses on functionality; however, adding usability testing would enhance the assessment of user interaction and overall gameplay experience. Testing will be carried out using a gameplay test to test whether there are obstacles to a feature that has been made (Table 3). In addition to testing gameplay, an analysis of the needs of the device to run the Odyssey Supply Game (Table 4).

Table 3. Gameplay Test

Testing	Test Results
Option Menu	All features in the options menu operated as expected, allowing real-time adjustment of audio and display settings.
Upgrades Skill Passive Menu	All skills can be upgraded to Max Level.
Quit Button	The exit button can function properly.
Play Button	Functioning well can start the game without any problems.
NPC Behaviour	Line of Sight functions well all NPCs can track and follow the player.
In-game Skill	All skills obtained can be used and can be improved properly.
Player Health	The life of the player can be reduced when in contact with the enemy, and can increase if the player has a skill.
Character Movement	The characters can move if you use the W button: to move up, A: to go to the left, S: to go down, and D: to go right.
Drop Monster	Every time defeating the monster will drop the experience to raise the player level.

3.5. Release

After a series of tests on the Supply Odyssey game, the final build was published on the Itch.io platform, where the game can be downloaded along with additional resources, including playthrough guides and gameplay tips. A performance analysis was conducted to assess the runtime efficiency of

Supply Odyssey, focusing on frame rate stability, resource usage, and the impact of the Line of Sight (LOS) system. Tests were carried out through simulations on representative low-end hardware profiles. Performance analysis on low-end hardware shows that although Supply Odyssey runs stably at the beginning, after around 10 minutes of continuous play the frame rate gradually drops and CPU usage can reach 100% due to intensive enemy activity and repeated Line of Sight (LOS) calculations.

Table 4. Minimum System Requirements

Device	Device Details
Processor	2 Ghz
RAM	2 GB
Hardisk Space	100 MB available space
GPU	GeForce 7600 GS (512 MB) or equivalent
Sound Card	DirectX Compatible Sound Card Operating System

4. Conclusion

This research successfully achieved its main objective, which was to design and develop Supply Odyssey, a 2D Action RPG survivor-like game using the Godot Engine and the Game Development Life Cycle (GDLC) method. Each phase of the GDLC — including initiation, design, production, testing, and release — was implemented systematically to ensure the game met its intended functionality and design objectives. The key finding of this research lies in the implementation of the Line of Sight (LoS) system, which enhances non-player character (NPC) behavior by making enemy detection more realistic and strategic. Testing results confirmed that core gameplay features, such as combat, upgrades, and skill systems, operated effectively under minimal hardware specifications. Performance testing confirmed Godot's efficiency and flexibility in handling 2D RPG mechanics, although prolonged gameplay revealed minor memory leaks that gradually increased resource usage and caused frame rate instability. For future studies, it is recommended to enhance the artificial intelligence (AI) module for NPC behavior, expand level design to include dynamic environments, and port the game to other platforms (e.g., Android or web-based systems) to reach a wider audience. Further optimization of the Godot Engine, particularly in memory management and resource allocation, is also suggested to improve long-term performance stability.

References

[1] L. Zhang, "Application of Artificial Intelligence Technology in Game NPC," *EUDL,* 2024.

[2] Tandicha Marchelputra Siswoko, Hanny Haryanto, Khafiizh Hastuti and Ristia Kadiasti, "Non-Playable Character (NPC) based on Behaviour Tree for Enhanced Immersive Experience in the Serious Game "Warik's Adventure"," *Journal of Applied Informatics and Computing,* vol. Vol 7, no. 2, p. 284–290, 2023.

[3] Nidhi Kulkarni, Iffah Aafreen and Aesha T Patel, "THE DEVELOPMENT OF MINIX: A 2D PIXEL ART GAME USING GODOT 4 ENGINE," *International Journal of Engineering Applied Sciences and Technology,* vol. Vol. 9, no. Issue 07, pp. Pages 43-48, 2024.

[4] Sujacka Retno, Zeny Arsya Fortilla and Ilmi Suciani Sinambela, "Developing the Console Dash: a 2D Adventure Game using Godot Game Engine," *Gameology And Multimedia Expert,* vol. Vol.1, no. No.1, pp. pp. 10-15, 2024.

[5] Leonard Winata, Muhammad Akbar Maulana and Joko Susilo, "Studi Perbandingan Pengembangan Game dalam GDScript dengan Godot dan C# dengan Unity," *Bit-Tech(Binary Digital -Technology),* vol. Vol.7, no. 3, 2025.

[6] T. M. Brogan, "Evaluating the efficiency of general purpose and specialized game engines for 2D games," *Purdue University Graduate School,* vol. V 1, p. 3, 2024.

 DOI: https://doi.org/10.52435/jaiit.v7i2.720

[7] Mahesh Ranaweera and Qusay H. Mahmoud, "Deep Reinforcement Learning with Godot Game Engine," vol. VOL 13, no. 5, 2024.

[8] Robert Kilijanek and Marek Miłosz, "Comparative Analysis of the Performance of Unity and Unreal Engine," *Journal of Computer Sciences Institute,* 2025.

[9] Li J Wang X and Zhao L, "Design and Implementation of 2D Role-Playing Games Using Unity Engine and C#," *International Journal of Advanced Computer Science and Applications,* vol. 115–122, p. 14 (2), 2023.

[10] Masoud Rahimi, Hadi Moradi, Abdol-hossein Vahabie and Hamed Kebriaei, "Continuous Reinforcement Learning-based Dynamic Difficulty Adjustment in a Visual Working Memory Game," *arXiv,* 2024.

[11] Julian Holfeld, M. Sc., "On the relevance of the Godot Engine in the indie game development industry," *arXiv,* p. 2401.01909v2 [cs.OH], 2024.

[12] Paul Williamson and Christopher Tubb, "Heatmap Weighted A* Algorithm for NPC Pathfinding and Graph Switching," *International Journal on Advances in Software,* vol. Vol 16, pp. 1-2, 2023.

[13] Rahman A and Pratama H, "The Effect of Visual Design on Player Engagement in 2D Action RPG Games," *Journal of Game Development Research,* vol. 5, no. 1, p. 33–45, 2024.

[14] Qingwei Mi and Tianhan Gao, "Improved Belgian AI Algorithm for Dynamic Management in Action Role-Playing Games," vol. Vol 12, no. 22, 2022.

[15] SteamDB, "SteamDB Action RPG Chart Release," SteamDB, 4 November 2025. [Online]. Available: https://steamdb.info/stats/releases/?tagid=4231.

[16] Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, Jiamang Wang and Bo Zheng, "CombatVLA: An Efficient Vision-Language-Action Model for Combat Tasks in 3D Action Role-Playing Games," *arXiv,* 2025.

[17] Yutong Yao, Ruifang Cui, Yi Li, Lu Zeng, Jianliang Jiang, N. Q. Li Dong, Diankun Gong, Guojian Yan, Weiyi Ma and Tiejun Liu, "Action Real-Time Strategy Gaming Experience Related to Enhanced Capacity of Visual Working Memory," *Frontiers,* vol. Vol 14, p. 333, 2020.

This page is intentionally left blank

 DOI: https://doi.org/10.52435/jaiit.v7i2.720