

# Optimization of Player Experience and Enemy AI using A\* Algorithm in Game

Dani Arifudin<sup>1\*</sup>, Michael Fransjaya<sup>2)</sup>, Yusif El Fakhry<sup>3)</sup>, Author Hikmalul A'la Syahrizaldy<sup>4)</sup>

<sup>1,2,3,4)</sup>Program Studi Teknologi Informasi, Universitas Amikom Purwokerto, Indonesia

<sup>1)</sup> daniarif@amikompurwokerto.ac.id, <sup>2)</sup> michizks000@gmail.com, <sup>3)</sup> yusifelfakhry@gmail.com, <sup>4)</sup> hikmalul0503@gmail.com

**Submitted** : Dec 26, 2024 | **Accepted** : Jan 31, 2025 | **Published** : Feb 3, 2025

**Abstract:** The gaming industry is rapidly evolving, where engaging and challenging gameplay has become a key factor in a game's success. Effective enemy intelligence can enhance challenges and enrich the player experience. This study aims to improve the player experience and enemy intelligence in the game Galang the EcoRescue through the implementation of the A-star (A\*) algorithm. A\* is a pathfinding algorithm that uses distance estimation to find the shortest path to a target by utilizing a heuristic function. This game was developed using the Unity Engine, with the implementation of the A\* algorithm to determine enemy movements and adapt their behavior according to the game's situation. Testing was conducted to ensure improvements in both the player experience and enemy intelligence. The results of the study show that the A\* algorithm successfully enhanced enemy intelligence by creating more realistic and adaptive movements in response to the player, ultimately providing more dynamic challenges and improving overall gameplay quality. This study utilized the Game Development Life Cycle (GDLC) method, covering the stages of initiation, pre-production, production, testing, beta, and release. The A\* algorithm has proven to optimize the player experience in Galang the EcoRescue.

**Keywords:** A\* Algorithm, Artificial Intelligence, Game, Gameplay, Pathfinding

## INTRODUCTION

The advancement of technology has made the role of games more than just entertainment. Today, games serve various functions, including educational media known as educational games (Widodo & Ahmad, 2017). Educational games utilize game elements to deliver learning materials, ranging from teaching specific subjects to enhancing learning motivation (Wijayanto & Siradj, 2017). Research on the use of game-based learning models concludes that it can reduce boredom in the learning process, making students more enthusiastic and actively participating in lessons. This method of learning is known as Game-Based Learning (Arifudin et al., 2022). With the integration of AI, games have become much more engaging and interactive.

The application of artificial intelligence (AI) in the development of modern games has significantly transformed the gaming experience. It presents new challenges for players, making games more dynamic and engaging. This development has influenced various fields of AI research, including computational intelligence, algorithms, machine learning, and combinatorial game theory (Fujita & Wu, 2012). A key objective in the development of captivating games is to create an interactive experience for players. This experience is shaped by various game elements, including narrative, challenges, graphics, sound, the timing of events and phenomena, and entities that directly interact with the players, whether they are opponents, allies, or other objects within the game (Westera et al., 2020). Games offer an immersive gameplay experience, often challenging players to uncover secrets and overcome obstacles on their journey. Recognizing the diversity of game genres is essential for understanding the complexity and variation in the overall gaming experience (Adams & Dormans, 2012).

This research focuses on games related to environmental cleanliness. With the growing awareness of environmental issues, the demand for educational games that promote environmental care has also increased. *Galang the EcoRescue* is an example of an educational game that combines adventure, brain-teasing concepts, and awareness of environmental problems caused by waste. With its unique gameplay, this game provides an enjoyable experience while offering valuable insights into the harmful impact of waste on the ecosystem.

In the design of this game, an algorithm is needed to enhance the dynamics of gameplay. This algorithm adjusts the difficulty and challenges based on the player's actions. Additionally, enemy AI is implemented in the gameplay to allow enemies to make strategic decisions when pursuing the player or navigating through complex environments. The algorithm that will be applied to meet the needs of this game is the A\* (A-Star) Algorithm. The A\* Algorithm is a solution technique for pathfinding problems in games (Foad et al., 2021). The A\* Algorithm

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

calculates the best direction to move and can trace the path back. Not only does it find a route between the starting point and the goal, but it also identifies the shortest and fastest path (Agung et al., 2022).

The use of the A\* Algorithm in this game enables enemies to make intelligent and adaptive decisions in their movement across the game map. By considering the target's position and obstacles, the A\* Algorithm helps enemies plan the shortest or most efficient route toward the player or a specific goal. This enhances the gameplay experience by allowing enemies to exhibit realistic behavior and respond to the player's actions, making the game more challenging and engaging. Therefore, the implementation of the A\* Algorithm not only improves the overall gaming experience but also enhances the enemy AI in *Galang the EcoRescue*.

## LITERATURE REVIEW

The gaming experience refers to the characteristics of playing a game (including game rules, storyline, and missions) and how players master the game, resulting in a memorable experience. There are three categories of methods for measuring the experience in a game (Mandasari & Pudjoatmodjo, 2019).

### Quality of a Product (Game System Experience)

To assess the quality of a game system, this is akin to testing a game product in general, such as stress testing, unit testing, and beta testing.

### Quality of Interaction Between Humans and Products (Individual Player Experience)

Assessing the quality of an individual's knowledge can be done using various sensors to observe player behavior, including eye tracking, psychophysiological player testing, game behavior metrics evaluation, qualitative interviews, and questionnaires.

### The Quality of Social Interaction, Temporal, Spatial, and Other Contexts

Assessing the quality of social interaction can be done by observing collaboration, competition, and communication between players in the game. The quality of temporal (time) context is evaluated through play duration and game pace, while the spatial context includes navigation, environmental immersion, and player interaction with the virtual world.

The following Figure 1 explains the three categories of methods for measuring the gaming experience.

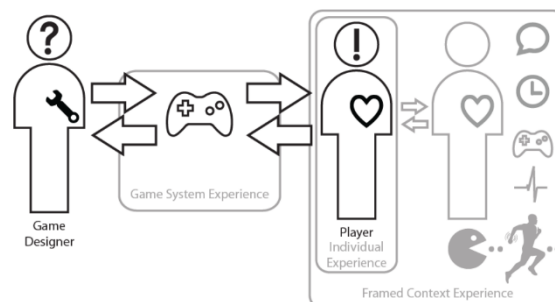


Figure 1. Gameplay Experience Categories

One approach to enhance the gaming experience is through the application of Artificial Intelligence (AI). In the context of games, AI is used to control computer characters so that they can interact with human players and serve as opponents in the game (Billy et al., 2017). The implementation of AI in enemies allows them to display more adaptive and challenging behavior for players. By utilizing various AI techniques such as reinforcement learning, pathfinding algorithms like A\*, or rule-based approaches, enemies can make smarter decisions when pursuing the player, avoiding obstacles, or even collaborating with other enemies. The goal of implementing AI in enemies is to increase player engagement and provide a more dynamic and engaging gaming experience.

To enhance the gaming experience, AI not only governs enemy behavior but also utilizes pathfinding to determine the optimal route in the game, creating more dynamic challenges. Pathfinding is the process of finding the fastest path from a starting point to a goal while avoiding obstacles along the way. Pathfinding can be divided into two types: undirected and directed. In undirected pathfinding, like an ant lost in a maze, the algorithm tries to find an exit randomly. There are two common approaches to solving undirected pathfinding problems: Breadth-First Search and Depth-First Search. In directed pathfinding, there is no random path search; before determining the best path, the algorithm first examines all adjacent nodes. Approaches such as Uniform Cost Search and Heuristic Search are used to evaluate these nodes before deciding on the best path (Iskandar et al., 2020).

One of the most effective directed pathfinding algorithms is the A\* (A-star) Algorithm, which is an enhancement of the Breadth-First Search (BFS) algorithm. Like BFS, A\* is guided by a heuristic function in the search for a solution. The difference between A\* and BFS lies in how the cost to reach the goal is calculated. A\*

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

not only considers the cost from the current state to the goal using a heuristic function, but it also estimates the cost from the initial state (i.e., the goal location) to the current state (i.e., the starting position). Therefore, if there is a cheaper path that offers the same solution from the goal's perspective, A\* will choose that path. The calculation used by the A\* algorithm follows the formula stated in Equation 1 (Octavian & Hermawan, 2023). A\* uses BFS and finds the least-cost path from the given initial node to the goal node. This algorithm uses a heuristic function that combines distance and cost (commonly denoted as  $f(x)$ ) to determine the order in which the search traverses nodes in the tree (Gede Wahyu Antara Dalem, 2018).

The notation used by the A\* Algorithm is as follows (Idayat & Handayani, 2022):

**g(n)**: The cost from the start to node n.

**h(n)**: The estimated cost from node n to the goal (heuristic).

**f(n)**: The total predicted cost

**f(n) = g(n) + h(n)**

In its implementation, the A\* Algorithm has several basic terminologies, including:

#### **Starting Point**

The starting point of the pathfinding search, typically the initial node.

#### **Nodes**

Nodes are representations of the pathfinding area in the form of small tiles. These nodes can be squares, circles, or triangles.

#### **A**

The node currently being processed in the shortest path search algorithm.

#### **Open List**

A storage place for nodes that can be accessed from the starting point or nodes that are currently being explored.

#### **Closed List**

A storage place for nodes before node A that are also part of the shortest path that has already been successfully found.

#### **Cost**

The value obtained by summing the values of each node in the shortest path from the starting point to node A, plus the estimated value of a node to the goal node.

#### **Unwalkable**

An attribute that indicates a node that cannot be traversed by the A\* algorithm.

In this research, the implementation of the A\* Algorithm will be applied through the development of the game *Galang the EcoRescue*, an environmental rescue game focused on waste. The game invites players to become heroes in saving the environment from the dangers of waste, which threatens the ecosystems of animals and plants, makes human living conditions unhealthy, and aims to raise awareness about the harmful impacts of waste on the environment. In this game, players will explore a world impacted by waste, starting from urban areas to remote natural ecosystems such as forests. Through a series of carefully designed levels, players will face puzzles that require creativity and strategic thinking to solve. The collection of items found during the adventure will be highly valuable, not only as an important aspect of the game but also as new knowledge that players can learn, available when successfully discovered.

## **METHOD**

The method underpinning the development of this game is the Game Development Life Cycle (GDLC). This method ensures that the game development process is carried out in a structured manner. Figure 2 below illustrates the stages of the GDLC method (Ramadan & Widyani, 2013).

\*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

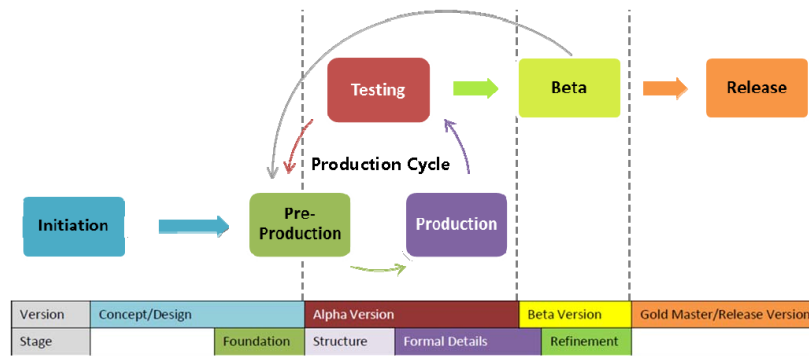


Figure 2. Game Development Life Cycle

**Initiation**

The initiation phase involves creating the initial game concept, defining gameplay, and identifying the target audience. At this stage, the expected outcome is a simple game concept and description. This process also includes determining the gameplay mechanics and conducting research on the necessary methods and algorithms. Additionally, this phase involves selecting the game engine, which in this case is Unity.

**Pre-production**

In this phase, the process of creating and revising the game takes place, along with the development of a game prototype. The game is developed while considering the game genre, gameplay, game mechanics, characters, difficulty levels, technical aspects, and game documentation. This phase ends once the revisions or design changes have been approved. The game prototype is created using simple assets, with the main focus on mechanics and gameplay. During this phase, a grid map is created as a tool for pathfinding within the game. This grid map consists of squares or nodes that form a map. The initial step in creating the grid is to generate nodes or squares arranged in a grid using a script.

**Production**

This process includes the creation of game assets and source code, including enhancing details, adding new features, improving overall performance, and fixing any remaining bugs in the game. Activities during this refinement phase aim to make the game more interactive, challenging, and easier to understand. At this stage, the game is enriched with additional assets, performance is enhanced, and bugs are fixed. This process also involves linking the A\* algorithm script with the prepared prefab by creating scripts in the prefab to link the variables between the A\* algorithm script and the prefab scripts.

**Testing**

This phase involves internal testing to evaluate the overall performance of the game. During this process, if there are bugs or failures, the causes and scenarios are identified and documented. Testing is also done to improve accessibility quality, which can be evaluated through behavioral observations during the test. If testers encounter difficulties in playing or understanding the game, this indicates that the game is not sufficiently accessible. The results of the testing phase will determine whether the game can proceed to the next phase or requires iteration within the production cycle.

**Beta**

In this phase, third-party testers or beta testers are involved. The testing methods used in beta testing are the same as those used in previous testing phases. During this stage, testers are asked to find bugs so that the game can be refined, and they are given the freedom to use the game as they wish. The results of the beta testing phase include bug reports and feedback from the testers. The beta phase will move to the next stage once the beta testing period ends or when the number of beta testers meets the established criteria.

**Release**

At this stage, the game has reached its final phase and is ready for release. The release process involves creating project documentation and planning for maintenance and expansion of the game. After the game is completed, the final step is to maintain the game and add additional features.

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

## RESULT

### Initiation

During the initiation phase, the initial concept of the game *Galang the EcoRescue* was developed, combining adventure elements with the goal of raising awareness about environmental issues. The main outcome of this phase was the creation of the gameplay concept, a simple description, and the identification of the target audience interested in challenging educational games. Research on the methods and algorithms required for game development was also conducted, with the A\* Algorithm selected as the optimal solution to enhance enemy AI in the game. Additionally, Unity Engine was chosen as the development platform due to its ability to efficiently support AI integration and algorithm implementation. This phase became a crucial foundation for the further development of the game.

### Pre-Production

In this phase, the process of creating and revising the game and building the game prototype was carried out. The game was developed by considering the game genre, gameplay, game mechanics, characters, difficulty levels, technical aspects, and game documentation. This phase concluded after the design revisions or changes were approved. The game prototype was created using simple assets, with the primary focus on mechanics and gameplay. Below, in Figure 3, are the temporary character movement assets, such as running, jumping, and idle animations using stickman assets, as well as environment assets like grass, trees, rocks, and more.

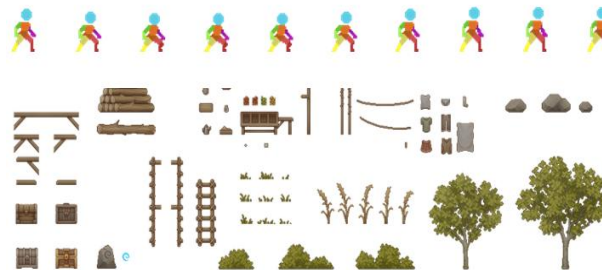


Figure 3. Character and environment assets

Then, at this stage, a grid map is created as a tool used for pathfinding in the game. This grid map consists of squares or nodes that form a map. The first step in creating the grid is to create the nodes or squares arranged on the grid using a script. The following, in Figure 4, is the pathfinding grid map.

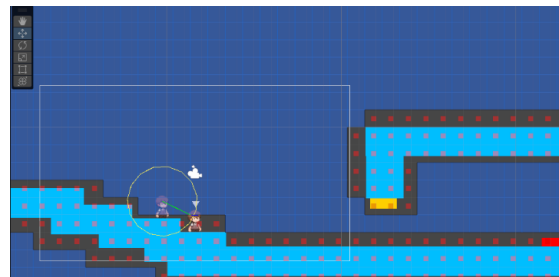


Figure 4. Pathfinding Grid Map

### Production

This process includes the creation of game assets and source code, such as improving details, adding new features, enhancing overall performance, and fixing any existing bugs in the game. The following, in Figure 5, shows the stage of character and environment asset development.

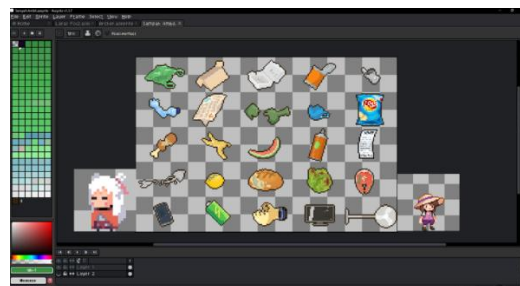


Figure 5. Asset Development

\*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

In the production stage, the refinement process aims to make the game more interactive, challenging, and easier to understand. During this stage, the game is enriched with additional assets, performance is improved, and bugs are fixed. Furthermore, at this stage, a diagram for implementing the A\* algorithm in pathfinding is created. Below, in Figure 6, is the diagram.

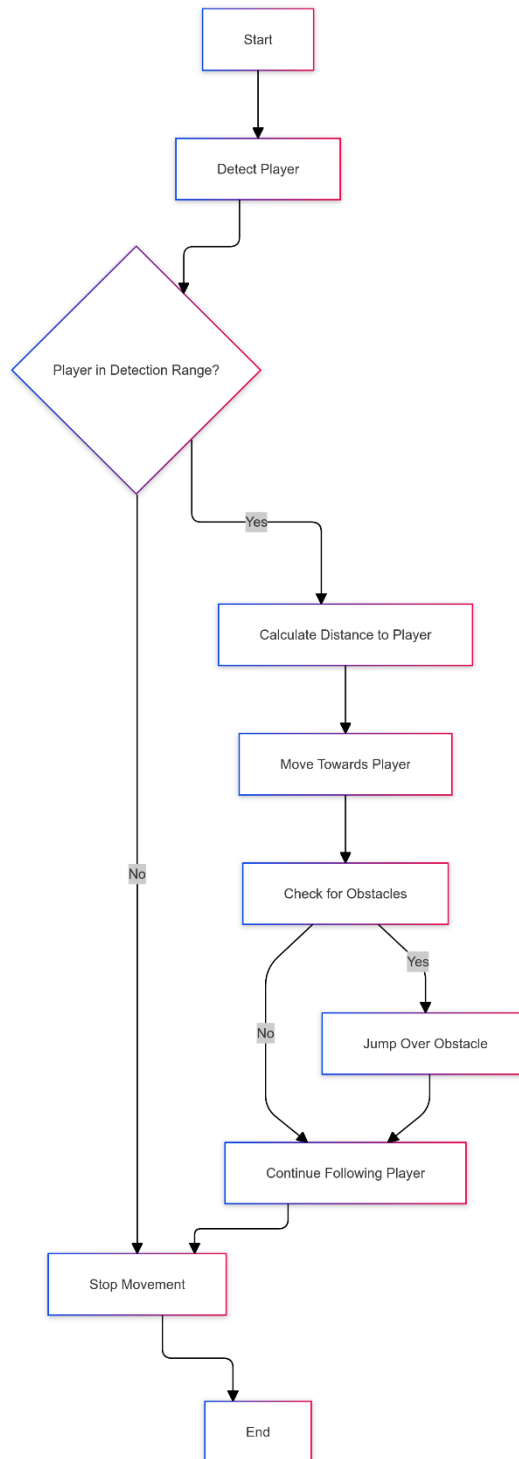


Figure 5. A\* Algorithm Implementation Diagram in Pathfinding

This process also involves connecting the A\* algorithm script with the prefabs that have been prepared, by creating a script on the prefab to link the variables between the A\* algorithm script and the script within the prefab. After the A\* algorithm implementation is complete, the next step is to create a flowchart that illustrates how the AI works effectively in the game. Below, in Figure 7, is the AI implementation flowchart for the game.

\*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

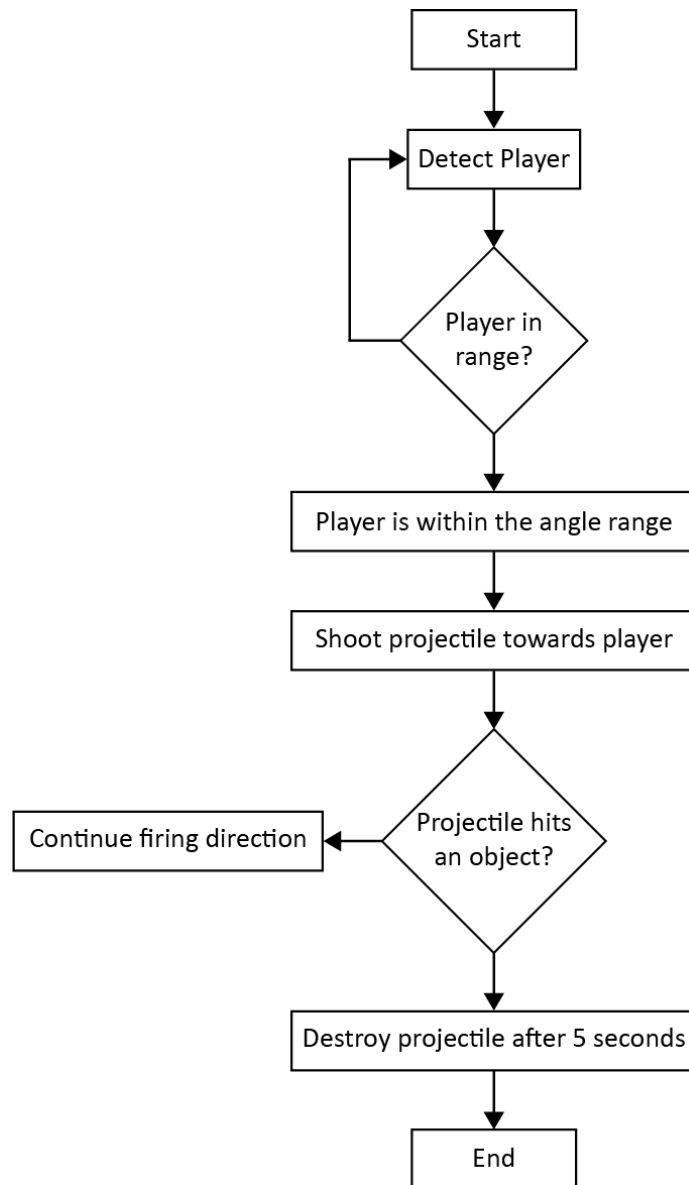


Figure 7. AI Implementation Diagram

**Testing**

The testing phase aims to evaluate the implementation of the A\* Algorithm in the game *Galang the EcoRescue*, ensuring that pathfinding and enemy AI work optimally. Various scenarios are tested, such as enemy movement in open and closed areas, obstacle avoidance, responses to changes in the player's position, and performance when facing multiple enemies. This testing is expected to provide insights into the performance of the A\* Algorithm and assist in optimizing the gameplay experience. Below, in Table 1, is the testing scenario.

Table 1. Testing Scenario

No.	Testing Scenario	Expected Outcome
1.	Pathfinding in Open Areas	Enemies can detect the player and select the closest path without obstacles.
2.	Pathfinding with Obstacles	Enemies detect obstacles and look for an alternative path.
3.	Enemies Chasing Player in Dynamic Areas	Enemies find the most efficient path through obstacles.
4.	Enemies Chasing Player with Changing Position	Enemies dynamically adjust their path when obstacles appear.
5.	Player Moves Position Quickly	Enemies adjust their path quickly when the player changes position.

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

6.	Enemies Stop Chasing Player When Out of Range	Enemies stop pursuing the player when the player moves out of range.
7.	Multiple Enemies	Some enemies chase the player without collisions.
8.	Enemies Chasing Player in Complex Conditions	Enemies move efficiently in larger maps with obstacles.
9.	Enemies Chasing Player with Performance Optimization	Game runs smoothly even with many enemies in larger areas.
10.	Path Optimization	Enemies choose the most optimal path to avoid dangers.

### Beta

The beta testing of the game *Galang the EcoRescue* involved several external testers who explored the game freely. Overall, the feedback from the testers was very positive, particularly regarding the responsive controls and dynamic gameplay. The testers praised the implementation of the A\* Algorithm, which successfully made enemy movement more dynamic. Performance issues at more complex levels were identified, and some minor bugs related to enemy pathfinding in certain areas were found. However, no major issues were identified that significantly impacted the gameplay experience. The results of the beta testing indicate that the game is in good condition, with only a few minor improvements needed before the final release stage. Feedback from the testers will be used to further optimize the gameplay experience before the release phase.

### Release

*Galang the EcoRescue* is ready for release after all bugs were fixed and the game optimized based on feedback from the beta testers. Project documentation, including technical and user guides, has been prepared. The development team has also devised a maintenance plan for updating the game post-release, as well as expansion plans to add new features and levels to maintain player engagement.

## DISCUSSIONS

In this study, the implementation of the A\* Algorithm in the game *Galang the EcoRescue* successfully enhanced enemy AI and optimized the pathfinding system. The test results show that the A\* Algorithm provided more realistic and efficient movement for enemies in navigating the map, particularly in open and closed areas. A comparison between the previous pathfinding method and the A\* Algorithm showed a significant improvement in terms of the speed of enemy movement and their ability to avoid obstacles.

One interesting finding in this study is that the enemies were able to quickly adapt to changes in the player's position, thereby increasing the dynamic nature of the gameplay. This was evident in tests conducted in various scenarios, including enemy movement in areas with many obstacles. However, at more complex levels, there were some performance issues that need to be addressed further. For instance, although the A\* Algorithm optimized paths, in some cases with many enemies, the execution speed slightly decreased.

In this context, the results can be compared with other methods, such as the Dijkstra algorithm or Breadth-First Search (BFS). The A\* Algorithm has an advantage in terms of efficiency, especially when paired with an appropriate heuristic. However, further comparisons with other algorithms could be conducted to evaluate whether there are better alternatives for this game.

Moreover, the implementation of the A\* Algorithm in this game not only enhanced enemy AI but also provided a more challenging and enjoyable gameplay experience for players. Beta testing results showed that testers provided highly positive feedback regarding the interactivity and responsiveness of the game controls.

Another new finding is that the use of artificial intelligence (AI) algorithms like A\* in educational games can enrich the learning experience for users, especially in raising awareness about environmental issues. Therefore, this research contributes significantly to the development of AI-based games that are not only entertaining but also educational.

Overall, while there are areas for improvement regarding performance and optimization, the results of this study indicate that the implementation of the A\* Algorithm has positively impacted both the quality of gameplay and enemy AI in *Galang the EcoRescue*.

### Future Work

For future research, it is recommended to explore the use of other artificial intelligence algorithms or a combination of different algorithms to further improve enemy AI and enrich other aspects of the game. This is expected to contribute further to the development of AI-based educational games.

\*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

## CONCLUSION

The findings of this research show that the implementation of the A\* Algorithm significantly enhanced the enemy AI in the game *Galang the EcoRescue*, resulting in an improved user gameplay experience. The enemies in the game are able to respond adaptively and efficiently to the environment and the player's actions, creating a more dynamic and complex challenge. The testing conducted shows that the A\* Algorithm is capable of optimizing the enemies' paths with more realistic and efficient movement, ultimately enhancing the overall quality of gameplay. The game development process, using the Game Development Life Cycle (GDLC) methodology, successfully created a game prototype that meets the criteria for interactive and educational gameplay, particularly in raising user awareness of environmental issues. The game has undergone both internal and external testing, which demonstrates that the A\* Algorithm implementation works optimally, with positive feedback from beta testers.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to the Research and Community Service Institute (LPPM) of Universitas Amikom Purwokerto for their full support during the execution of this research, which allowed the study to be completed on time.

## REFERENCES

- Adams, E., & Dormans, J. (2012). Game Mechanics: Advanced Game Design. In *Game mechanics : advanced game design*.
- Agung, E. G., Eridani, D., & Fauzi, A. (2022). Implementasi metode pathfinding dengan algoritma a\* pada game rogue-like menggunakan unity. *Indonesia Journal on Computing (Indo-JC)*, 1(3), 81–89. <https://doi.org/10.34818/indojc.2022.7.3.677>
- Arifudin, D., Suliswaningsih, S., Pramesti, D., & Heryanti, L. (2022). Implementasi Game Design Document Pada Perancangan Game Based Learning. *CogITo Smart Journal*, 8(2), 385–397. <https://doi.org/10.31154/cogito.v8i2.431.385-397>
- Billy, Kuswardayan Imam, & Wijayanti Nurul Khotimah. (2017). Implementasi Artificial Intelligence pada Game. *Jurnal Teknik Pomits*, 6(2)(2), A640–A645. <https://ejurnal.its.ac.id/index.php/teknik/article/download/25151/4662>
- Foad, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., & Gunawan, E. (2021). A Systematic Literature Review of A\*Pathfinding. *Procedia Computer Science*, 179(2020), 507–514. <https://doi.org/10.1016/j.procs.2021.01.034>
- Fujita, H., & Wu, I. C. (2012). A special issue on artificial intelligence in computer games: AICG. *Knowledge-Based Systems*, 34, 1–2. <https://doi.org/10.1016/j.knosys.2012.05.014>
- Gede Wahyu Antara Dalem, I. B. (2018). Penerapan Algoritma A\* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 1(1), 41–47. <https://doi.org/10.31598/jurnalresistor.v1i1.253>
- Idayat, R., & Handayani, I. (2022). Penerapan Algoritma A\*Star Menggunakan Graph Untuk Menentukan Rute Terpendek Berbasis Web. *Pendidikan Dan Informatika*, 1(1), 7–14. <https://journal.mediapublikasi.id/index.php/manekin>
- Iskandar, U. A. S., Diah, N. M., & Ismail, M. (2020). Identifying Artificial Intelligence Pathfinding Algorithms for Platformer Games. *2020 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2020 - Proceedings, June*, 74–80. <https://doi.org/10.1109/I2CACIS49202.2020.9140177>
- Mandasari, R. I. M., & Pudjoatmodjo, B. (2019). Player Experience Measurements Using Immerse Experience Questionnaire on Pewayangan Game. *IJAIT (International Journal of Applied Information Technology)*, 3(01), 10. <https://doi.org/10.25124/ijait.v3i01.1609>
- Octavian, F., & Hermawan, L. (2023). Penerapan Algoritma Pathfinding A\* dalam Game Dual Legacy berbasis Android. *Jurnal Buana Informatika*, 14(01), 20–29. <https://doi.org/10.24002/jbi.v14i01.6928>
- Ramadan, R., & Widyani, Y. (2013). Game development life cycle guidelines. *2013 International Conference on Advanced Computer Science and Information Systems, ICACIS 2013*, 95–100. <https://doi.org/10.1109/ICACIS.2013.6761558>
- Westera, W., Prada, R., Mascarenhas, S., Santos, P. A., Dias, J., Guimarães, M., Georgiadis, K., Nyamsuren, E., Bahreini, K., Yumak, Z., Christyowidiasmoro, C., Dascalu, M., Gutu-Robu, G., & Ruseti, S. (2020). Artificial intelligence moving serious gaming: Presenting reusable game AI components. *Education and Information Technologies*, 25(1), 351–380. <https://doi.org/10.1007/s10639-019-09968-2>
- Widodo, W., & Ahmad, I. (2017). Penerapan Algoritma A Star (A\*) pada Game Petualangan Labirin Berbasis Android. *Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, 3(2), 57–63. <https://doi.org/10.23917/khif.v3i2.5221>
- Wijayanto, P. W., & Siradj, Y. (2017). The Educational Game “Indonesian Tribes” for the Kindergarten Students.

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

---

*IJPTE : International Journal of Pedagogy and Teacher Education*, 1(1), 27–36.  
<https://doi.org/10.20961/ijpte.v1i1.8456>

\*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.