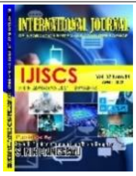

**IJISCS (International Journal of Information System and Computer Science)**
IJISCS, Volume 7, Number 3, December 2023
E ISSN: 2598-246X; P ISSN: 2598-0793, pp. 252-263
**Accredited SINTA 4 Number 79/E/KPT/2023**
https://jurnal.ftikomibn.ac.id/index.php/ijiscs


# OPTIMIZING RPG PATHFINDING A HYBRID APPROACH FOR STATIC AND DYNAMIC OBSTACLE AVOIDANCE


**Renaldy Ari Gunawan[1], Wirawan Istiono[2]***
[1,2]Depaetement of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1,2]Jalan Scientia Boulevard Gading, Curug Sangereng, Serpong, Kabupaten Tangerang, Banten

*Corresponding author
wirawan.istiono@umn.ac.id:[2]



**Article history**:
Received September 11, 2023
Revised October 31, 2023
Accepted December 14, 2023

**Keywords:**
A-star;
Dynamic Pathfinding;
Non-Player Character;
Pathfinding;
Role-Playing Game.



**Abstract**
Playing a role the popular game genre known as "RPG" employs a pathfinding algorithm in its operation. The process of discovering a path from one place to another is called pathfinding. Pathfinding algorithms come in a variety of forms and are varyingly efficient. One of these methods is the A-star algorithm, which is frequently used in different games to determine routes and has actually been evaluated for efficiency. An additional algorithm to consider is the dynamic pathfinding algorithm, which is quite tiny for reference and not well-known. Combining these two techniques can result in an algorithm that avoids both static and dynamic barriers in addition to finding the fastest path. The two algorithms will be combined in this research, and two static and dynamic obstacle scenarios will be used for testing. According to the study's findings, the combination of these two algorithms enables the algorithm to avoid both static and dynamic obstacles. However, it should be noted that in the event of a dynamic obstacle, the Non-Player Character (NPC) will need to travel 3.04 seconds longer to reach their destination because they will need to find an alternate route.


## 1.0 INTRODUCTION

A game is a form of recreational activity or amusement that is commonly utilized by individuals as a means of leisure during their available time [1]. The game, upon examination, encompasses numerous categories and genres, such as RPG (Role-Playing games), MOBA (Multiplayer Online Battle Arena), Racing games, and other more [1], [2]. The user's text does not contain any information. Among the assortment of features provided by the game, the inclusion of an Open World feature stands out as a particularly captivating aspect [3], [4]. The notion of an Open World-themed game involves the creation of game levels that offer players the opportunity to navigate and investigate their surroundings without constraints, hence granting them autonomy in devising strategies to accomplish objectives [5]. The inclusion of an open map element is prevalent in various game genres, including Role-Playing games [6]. The necessity for an expansive landscape in RPG games arises from the need to accommodate the primary player's exploration of the diverse features offered during gameplay [7], [8].

The Role-Playing Games (RPG) genre encompasses interactive digital entertainment experiences wherein participants have the opportunity to embody characters inside a simulated environment, sometimes set in a science fiction context [7], [9]. According to a poll conducted by AgateStudio, a prominent gaming studio in Indonesia, among a sample size of 1,200 players, it has been observed that the role-playing game (RPG) genre is currently the most favored among Indonesian gamers. According to the findings of a survey, RPG genre games have emerged as the most preferred games, garnering a significant majority of 46% [10]. The role-playing game (RPG) also has a variety of supportive characters known as Non-player Characters (NPCs), which are controlled by artificial intelligence (AI) [11], [12]. Non-player characters (NPCs) throughout many game genres serve the purpose of assisting players in attaining the ultimate objective of the game [13]. However, NPCs can also be strategically employed to enhance the competitive nature and immersive experience of the game. Within the context of this role-playing game (RPG) genre, non-playable characters (NPCs) possess a distinct functionality aimed at aiding the primary protagonist in attaining specific objectives or destinations. Furthermore, these NPCs are designed to be interacted with in a manner that emulates real-world interactions [14], [15].

During the course of their evolution, role-playing games (RPGs) have exhibited a multitude of characteristics that can be constructed. One notable aspect entails facilitating the primary player's navigation to a specific location without necessitating physical movement of their in-game character. Typically, this concept is employed in the context of companion animals or non-player characters (NPCs) that are capable of assisting the primary player. In situations where the desired destination is located at a considerable distance, the primary user has the option to activate the idle-travel function. This feature enables a companion creature or non-player character (NPC) to facilitate autonomous travel on behalf of the primary user, directing them towards a preselected location and identifying the most efficient way to reach such destination. One possible approach for implementing the algorithm suitable for this particular feature is to utilize the A* algorithm [16]. This algorithm is designed to efficiently determine the optimal path on a given track, prioritizing speed and efficiency. The A-star algorithm is described as being both straightforward to build and highly efficient. The effectiveness of the A-star algorithm for static barrier situations has been demonstrated in prior studies. However, in the presence of dynamic impediments, another method is required to identify and account for these obstacles [17]. This approach is commonly referred to as the dynamic pathfinding algorithm. The Dynamic Pathfinding algorithm bears resemblance to the A-Star method, with a notable distinction being its ability to identify and account for dynamic impediments encountered along its traversed path.

Based on the aforementioned description, this study aims to introduce a novel approach by integrating the Dynamic Pathfinding (DPA) algorithm with the A-star method for Non-Player Characters (NPCs) in the context of Role-Playing Games (RPGs) [18]. The utilization of these two algorithms is anticipated to enable non-player characters (NPCs) to effectively identify and navigate around dynamic barriers. This functionality aims to prevent NPCs from becoming immobilized at specific locations, hence enhancing the overall realism of the game.

## 2.0 THEORETICAL

### 2.1 ALGORITMA A STAR

The A* algorithm is a heuristic search algorithm that determines the fastest route by measuring the heuristic distance between specific points, although the search process itself depends on how the A* graph is represented in the field [16]. The A* algorithm initially searches for a general route, followed by a comparison and selection of the fastest among them. The illustration of the A* algorithm may be observed in Figure 1.

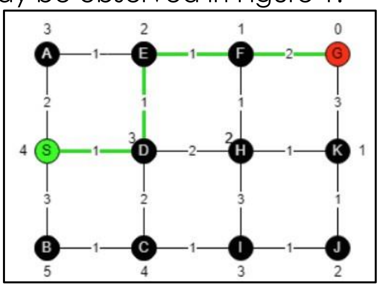

Figure 1. A Star Algorithm Demonstration

The A* method has been empirically demonstrated to effectively address the problem of finding the fastest route, but with the limitation of not accounting for dynamic barriers. When encountering a dynamic obstacle, this algorithm will enforce a specific route that may afterwards result in unintended visual glitches within the game. The A-star algorithm is extensively employed in diverse game genres, including Racing Games [19], [20]. An additional illustration pertains to a game that necessitates traversing a substantial distance between two points, hence necessitating the identification of the most expeditious route to ensure optimal efficiency of gameplay. Within the realm of RPG games, this algorithm is commonly employed to govern non-playable characters (NPCs) who inhabit the virtual world environment. These NPCs may include entities such as cars, which can be interacted with by the main player to facilitate expedited movement between various points within the virtual world [21].

### 2.2. DYNAMIC PATHFINDING

The dynamic pathfinding algorithm is a route search technique that shares similarities with the A-star algorithm. However, it possesses the added capability of detecting dynamic barriers inside the virtual environment. When the algorithm is transformed into the A-star algorithm, it will confront various challenges, including the need for NPC or agent information that includes details about the virtual world's structure through which it will navigate [22]. Based on the available data, the non-player character (NPC) possesses the ability to analyze the intricate details of the virtual world, deconstructing it into discrete components and identifying individual pathways. Conversely, in the event that the non-player character (NPC) lacks prior knowledge on the virtual world's environment, the algorithm in question would be unable to operate with optimal efficiency [23], [24]. In summary, the method being referred to is the A-star algorithm that has been adapted to incorporate the capability of identifying and circumventing dynamic barriers inside the virtual environment, so enabling the non-player character (NPC) or agent to navigate around them.

## 3.0 METHODOLOGY

Following are the research phases for devising and developing a 3D RPG. The first step in designing and developing this RPG game is to seek for references and study all relevant material on RPG game development and the algorithms used. During the design phase, flowcharts and the game's basic interface will be created and used as design references. When the game enters the game development phase. Then, proceed with game creation, where at this point the game has begun to be constructed using the Unity Engine and the previously created flowchart serves as a guide. C sharp is the programming language used, and the A-star Algorithm and Dynamic Pathfinding will be implemented for certain NPC entities.

Once the game development process is complete, the subsequent phase involves doing thorough testing to identify and rectify any potential bugs or issues that may arise. At this juncture, an examination will be conducted to ascertain the efficacy of the algorithm implementation applied to the non-player character (NPC) in accommodating the dynamic barriers presented in the map. The subsequent phase entails evaluation, during which an assessment will be conducted to determine the feasibility of playing the RPG game and utilizing non-player characters (NPCs) to assist the protagonist in attaining the designated objective or destination. The concluding phase of this research entails the creation of a comprehensive report encompassing the implementation of the A-star algorithm and the Dynamic Pathfinding algorithm within the context of an RPG-themed game. This report will also include a detailed elucidation of how these two algorithms operate on a non-player character (NPC), as well as an inventory of all the assets employed throughout the development of the RPG game.

Figure 1 is a flowchart illustrating the operational process of route finding within the game. When utilizing the idle travel functionality, the algorithm will automatically seek out the optimal route that incurs the least amount of expenses. In the event of encountering dynamic impediments that impede the character's trajectory, the dynamic pathfinding algorithm will effectively operate and identify these dynamic barriers. Upon successful detection, the dynamic pathfinding algorithm will assume control of the route and proceed to search for an alternative path in order to circumvent the dynamic obstacles. In contrast, in the absence of dynamic impediments, the default route search for the A-star method will be employed.

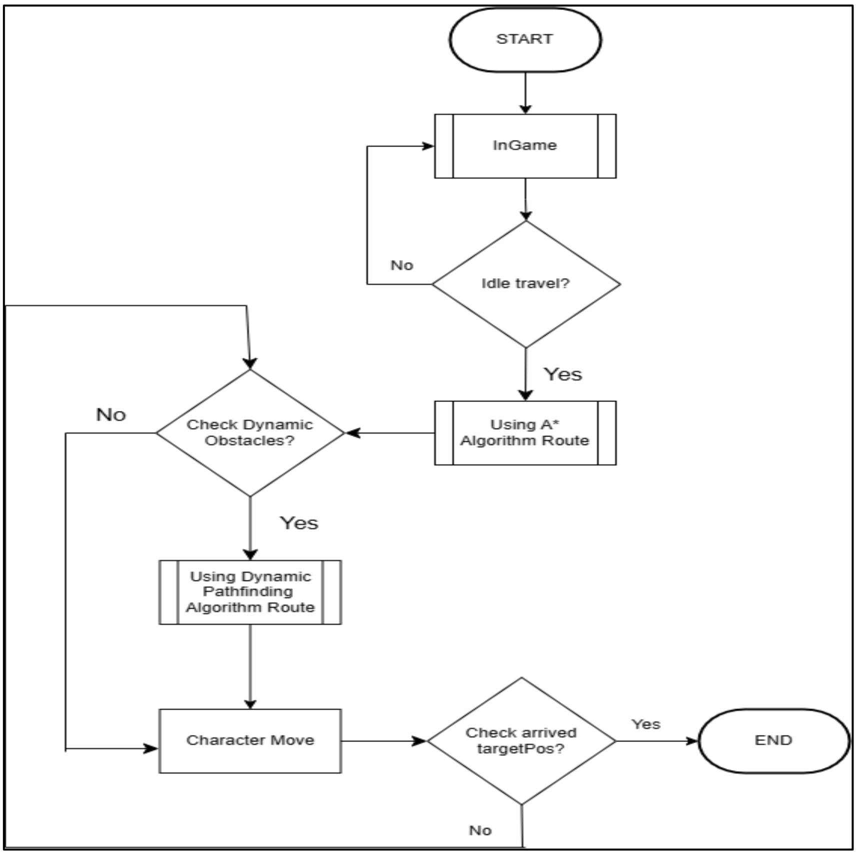


Figure 1. Flowchart Bear Pathfinding Route

A schematic of the A* Pathfinding algorithm is depicted in Figure 2. The algorithm begins by initializing the Lists openSet and closedSet, followed by initializing n, populating n's value with startPos, and adding n to the openSet.

$$f(n) = g(n) + h(n) \tag{3.1}$$

f(n) = accumulated distance between destination points and the heuristic distance.
g(n) = the original distance between the starting point to the final destination point.
h(n) = the heuristic distance from the starting point to the destination.

Calculate FCost using the formula f(n) = g(n) + h(n). where f(n) is the FCost, g(n) is the distance from the startPos Node to the destination Node, and h(n) is the heuristic value, the estimated distance from Node n to the ultimate target. Then, move Node n from openSet to closedSet; this indicates that the Node has been incorporated into the path that will be utilized. After that, the algorithm will look for Node n with the smallest f(n) value. The next step is to determine if n is the targetPos. If n is not the targetPos, the algorithm will continue by looking for n that is not in the closedSet, adding Node n to the openSet, calculating FCost, and then moving Node n from the openSet to the closedSet. If n is targetPos, then the algorithm has been executed successfully and the closest path has been constructed by tracing the existing closedSet.

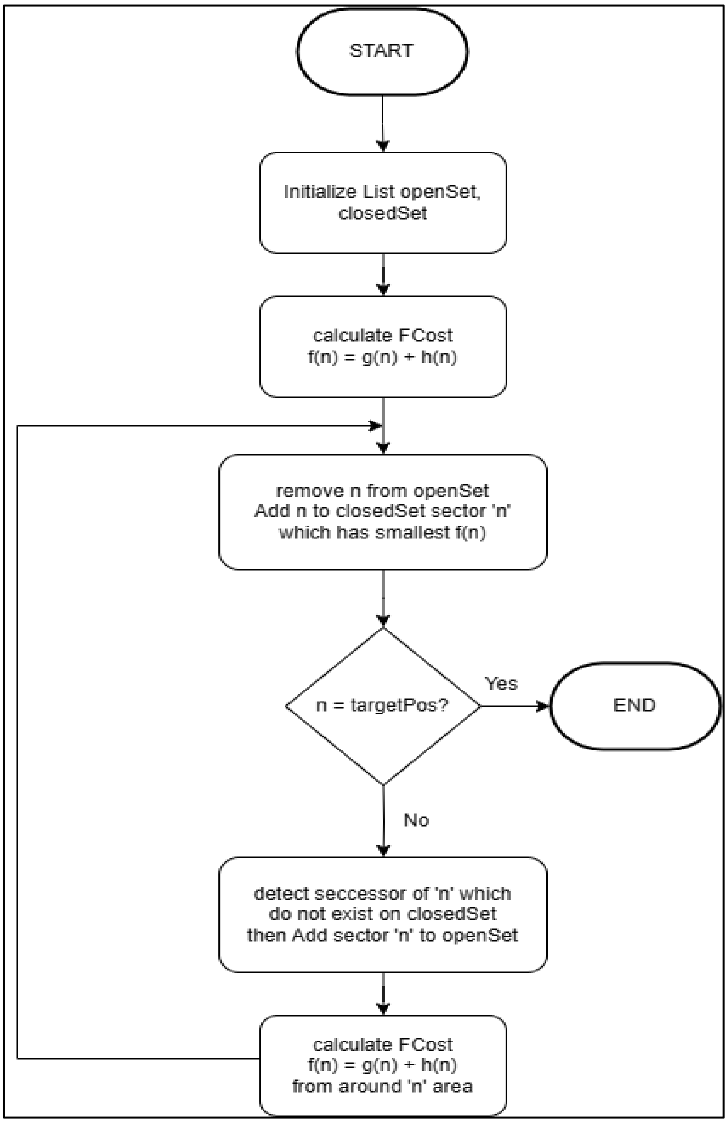

Figure 2. Flowchart Using A-star ALgorithm Route

The flow of the dynamic pathfinding algorithm is depicted in Figure 3. The algorithm in question operates in a similar manner to the A-star algorithm, as it employs the ideas and logic of the A-star algorithm in its implementation. One notable disparity lies in the capability to identify and evade dynamic barriers that manifest throughout the NPC's progression towards the target position. Subsequently, the node denoted as n representing the dynamic obstacle will be transferred from the openSet to the closedSet. Subsequently, the subsequent phase mirrors that of the A-star method, wherein the determination of the path to be traversed is recalculated by identifying the minimum f(n) value within the vicinity of area n. Once the value of n reaches the intended position, the character will come to a halt, and the algorithm will execute successfully in order to navigate around dynamic obstacles present in the map.

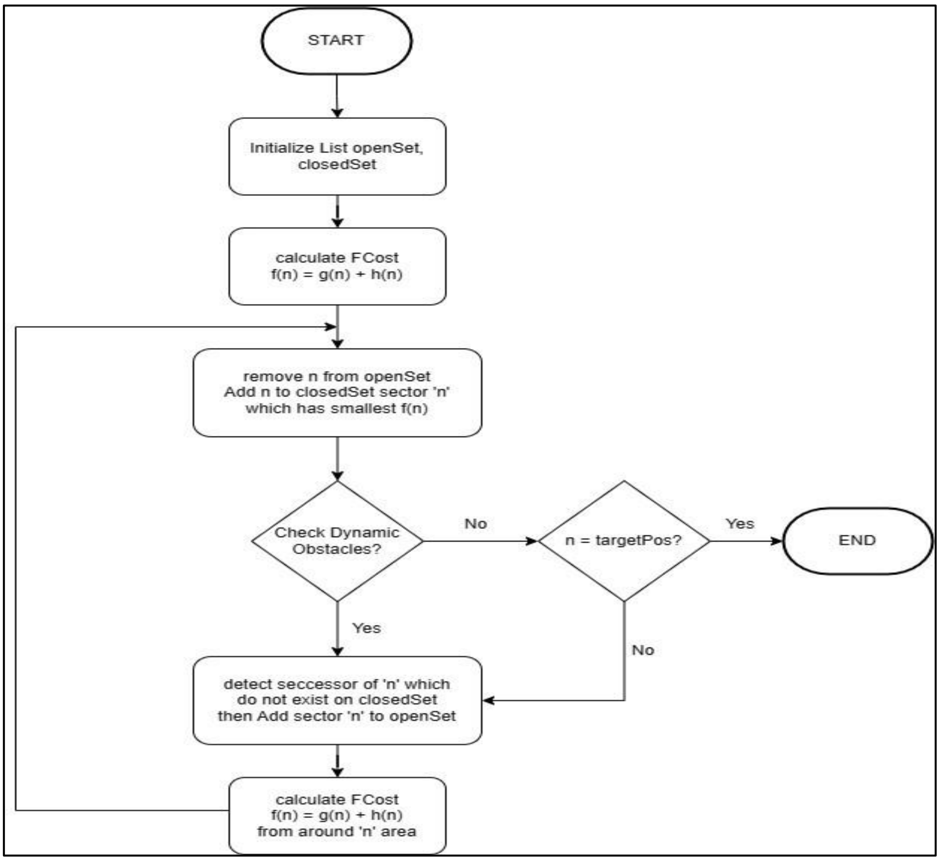

Figure 3. Flowchart Using Dynamic Pathfinding Algorithm Route

The flow of gameplay in this RPG game commences upon the player's initiation of the desired scene, as depicted in Figure 4. Players have the ability to halt gameplay at their discretion, regardless of their current location inside the virtual environment, as long as they remain within the confines of the active in-game setting. When a player initiates a pause, all ongoing operations within the game will temporarily cease, and the pause menu will be displayed. The game interface includes a menu option for exiting the game. To resume gameplay, the primary player can hit the Tab button once more. Subsequently, the game will additionally verify whether the non-playable character (NPC) who assigns a task has not been engaged with. In such a scenario, the sword item that has to be acquired will not be assigned the value of "Active = true" and will remain in the state of "false".

Once the non-player character (NPC) quest has been initiated, the sword item object will be activated by setting its Active attribute to true. This will enable the player to interact with the object and take it. In order to reach the designated location where the item is located, which can be considered a considerable distance away, the primary player has the option to utilize the "travel with bear" functionality. This feature allows the primary player to subsequently track the movement trajectory of the bear companion, which has been incorporated into the route search process through a combination of the A-star algorithm and the dynamic pathfinding algorithm. The utilization of the "travel with bear" functionality is contingent upon the primary player's proximity to their companion bear, followed by the activation of the "M" button. This action will prompt the display of the minimap, enabling the selection of the desired location. In addition, a verification process will be conducted to ascertain whether the pet bear has reached its intended location or is still in transit. Consequently, players will have the option to modify their destination without necessitating a wait for the pet bear to complete its journey. During the course of journey, the primary participant has the ability to revert back to the original camera perspective by activating the Escape key and thus discontinuing the minimap display. Upon arrival, it is advisable to verify the presence of the primary participant in the vicinity of the domesticated bear before proceeding to activate the "travel with bear" functionality. Nevertheless, in the event that you find yourself beyond the confines of the pet collider, this functionality will be rendered inoperative, hence necessitating the manual execution of your expedition by means of the default movement control. Subsequently, the initiation of the quest line might be instigated by one of the non-playable characters (NPCs) situated within the urban area. Upon the player's arrival at the collider, initiation of the quest can be achieved by the player's activation of the "E" key on the keyboard. Subsequently, two game objects will have their setActive property set to true. The aforementioned game entities are located within a specific sector on the map and are acquired by the primary player through the utilization of the "E" button. The confirmation of successful item collection can be observed in the subsequent display of the game text, which presents the cumulative count of collected things alongside the designated target. A notice will be triggered to indicate the completion of the quest and allow players to explore the game map region without any restrictions, if the quest item obtained or points reach a value of 2.

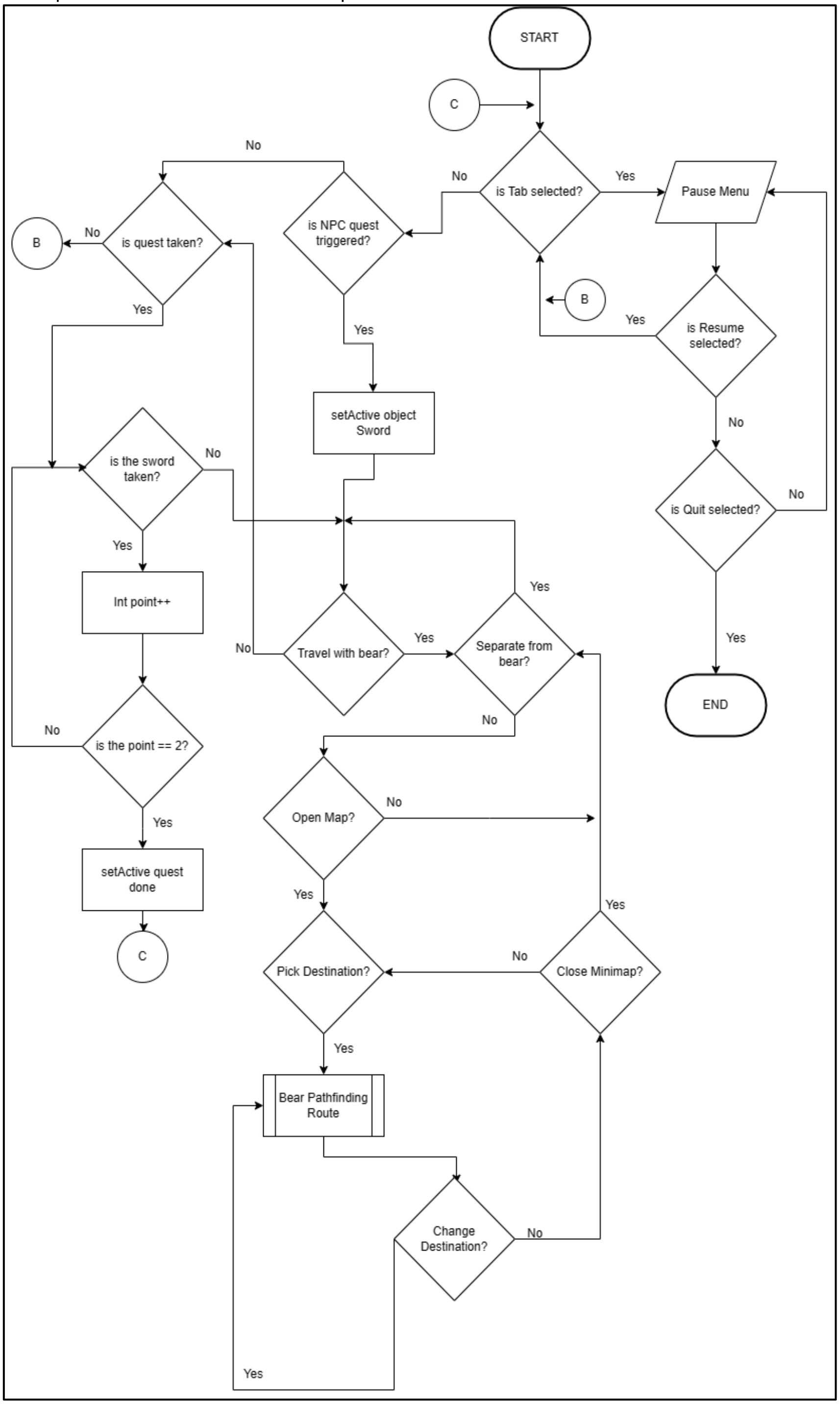


Figure 4. Ingame Flowchart

## 4.0 RESULT

The test display depicted in Figure 5 illustrates the experimental setup for scenario 1. In this particular scenario, the focus is solely on evaluating the performance in the presence of static impediments, with no inclusion of dynamic obstacles. The designated location for the placement of dynamic barriers in scenario 2 is shown by the red dot. The Tower and House buttons located in the lower right corner of the screen serve the purpose of activating dynamic impediments that are exclusively generated during the execution of the scene.

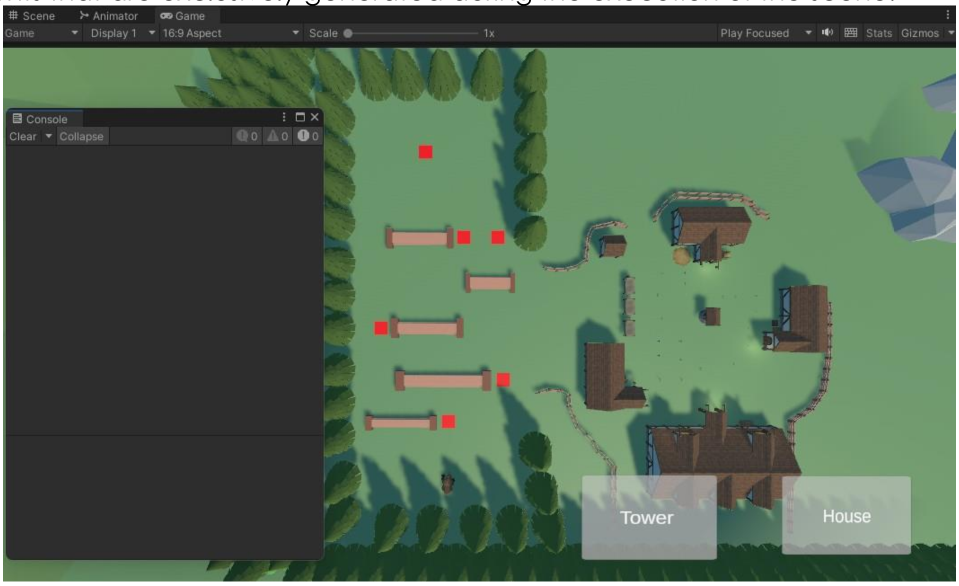


Figure 5. View of the pathfinding algorithm test

The outcome depicted in Figure 6 is the outcome of a route search, which is obtained by the utilization of both the A-star and Dynamic Pathfinding algorithms. The designated path will be utilized by the unit, which will then adhere to the computed route.

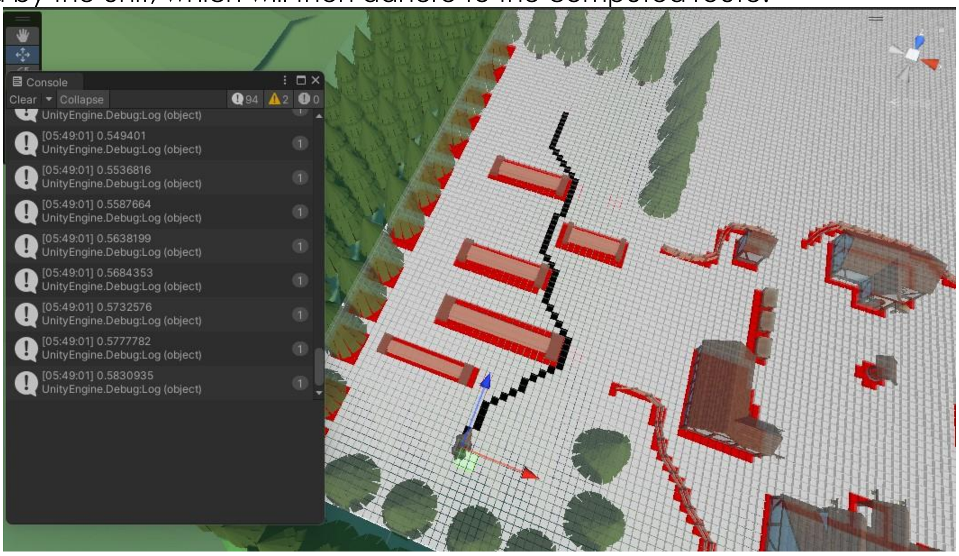


Figure 6. Dynamic obstacle-free pathfinding route search display

The data presented in Figure 7 illustrates the duration required for the unit to reach its designated location in the absence of any dynamic impediments within the mapped environment. The initial test had a recorded time of 15.14594 seconds on the console.

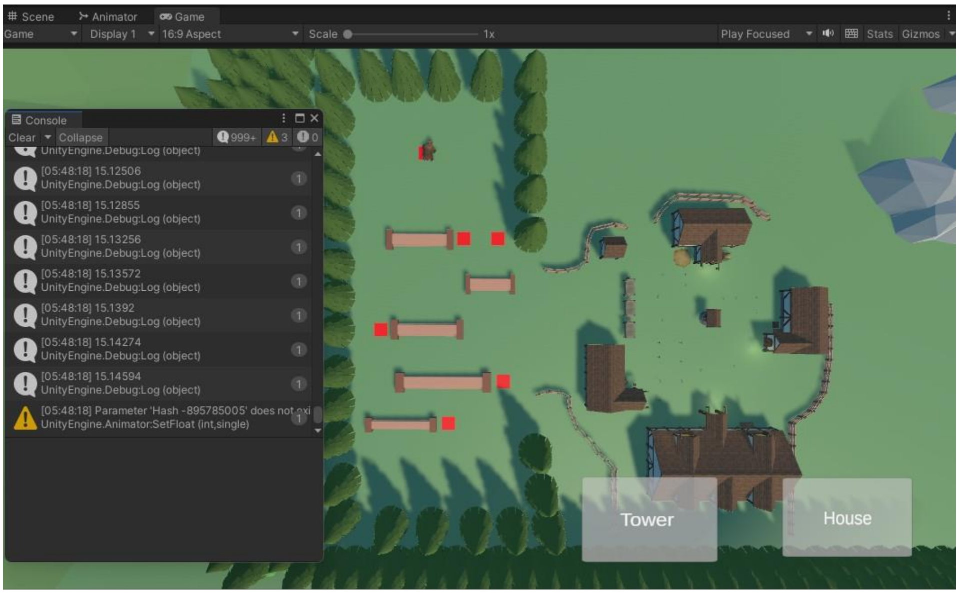


Figure 7. Display of the results of the time the unit arrives at the target position without dynamic obstacles

In the second scenario, a variety of dynamic obstacles are generated by the utilization of the tower and house buttons that are made available. Figure 8 illustrates the algorithm's search for an alternative route to circumvent dynamic impediments that were not previously present along the trajectory. The program will seek to identify the optimal and most efficient alternate route in the event that the default path is obstructed by unforeseen impediments.

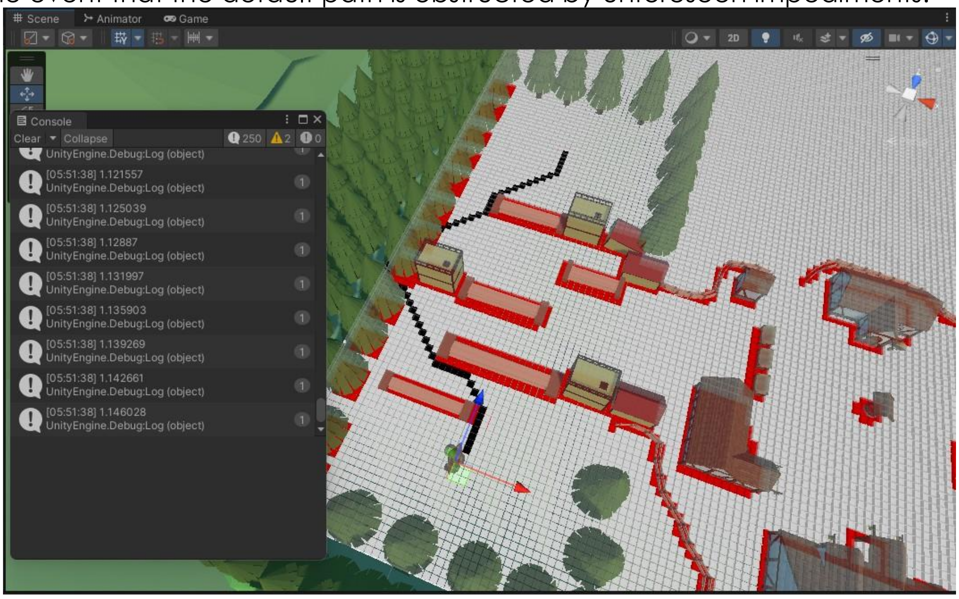


Figure 8. Route search display when faced with dynamic obstacles

The findings of the unit's response time to dynamic barriers and its ability to search for alternate paths when its original path is obstructed are depicted in Figure 9. The results of the second test revealed that the duration required to reach the identical target node was extended, namely lasting 18.03198 seconds.

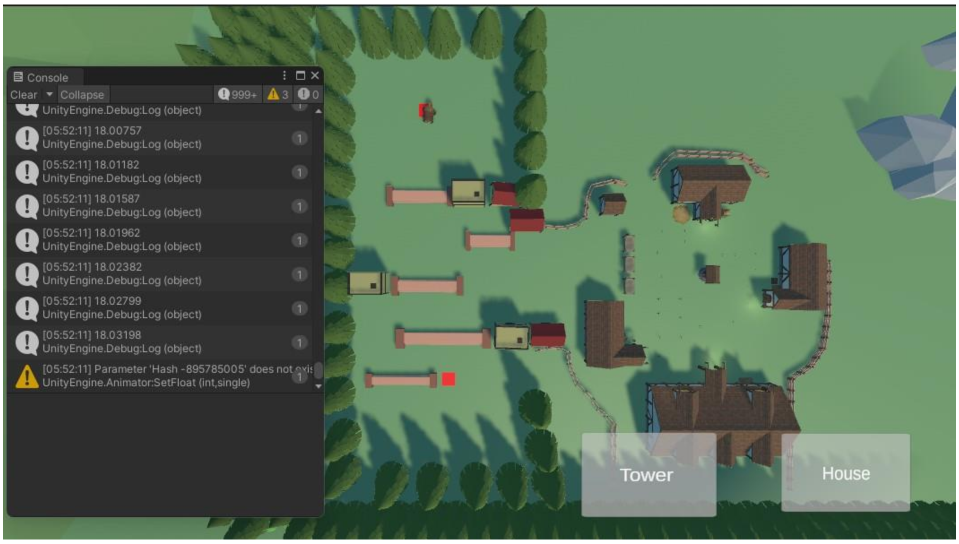

Figure 9. Displays the results of the time the unit arrives at the target position in the presence of dynamic obstacles

The algorithm underwent a total of five iterations in each scenario, with consistent positioning of both static and dynamic obstacles. The duration of travel is measured using a timer system implemented in Unity, and the resulting data is promptly presented in real-time on the editor interface. Table 1 displays the test table pertaining to scenario 1 and scenario 2.

Tabel 1. Dynamic no-obstacle travel time testing table

| Trials | Travel time (second) |
|---|---|
| 1 | 14, 98988 |
| 2 | 14, 98752 |
| 3 | 14, 98908 |
| 4 | 15, 1979 |
| 5 | 14, 99437 |

The findings presented in Table 1 illustrate the outcomes of the trip time analysis conducted for scenario 1 of route search. This analysis employed a hybrid approach that combined the A-star algorithm with dynamic pathfinding techniques. Notably, the analysis excluded the consideration of dynamic obstacles present on the map. Based on the results of the five conducted experiments, the mean travel time was calculated to be 15.03175 seconds.

Tabel 2. Travel time testing table with dynamic obstacles

| Trials | Travel time (second) |
|---|---|
| 1 | 18, 04538 |
| 2 | 18, 16537 |
| 3 | 18, 17768 |
| 4 | 17, 82718 |
| 5 | 18, 16273 |

The findings presented in Table 2 depict the trip time outcomes obtained from scenario 2 of the route search process. This particular scenario involved the utilization of a hybrid approach, combining the A-star algorithm with dynamic pathfinding techniques. The objective was to account for the presence of dynamic obstacles within the map. Based on the results of the five conducted trials, the mean travel time was determined to be 18.075668 seconds. When comparing the values in table 1, the average time difference for units to reach the target goal may be calculated as 18.075668 - 15.03175, resulting in a difference of 3.043918 seconds. The findings of this study demonstrate that the inclusion of dynamic barriers within the map leads to an increase in the time required for the unit to reach its destination. This delay occurs due to the necessity of the unit to search for an alternative route, deviating from the initially given path.

## 5.0 CONCLUSION

The findings of the conducted research indicate that the integration of the A-star and Dynamic Pathfinding algorithms has proven to be effective in facilitating the movement of Non-Player Characters within the RPG game. This implementation demonstrates the capability to identify both stationary and moving obstacles present in the game's map. The experimental evaluation of the method reveals that the inclusion of dynamic obstacles within the map leads to an increase in the journey duration for non-player character (NPC) units to reach their intended destination. This further demonstrates the capability of the dynamic pathfinding algorithm to identify impediments and persistently search for the most optimal alternate route inside the desired path. The mean time difference observed between tests conducted with dynamic obstacles present and tests conducted without dynamic obstacles is 3.043918 seconds.

## REFERENCES


[1] C. Potard *et al.*, "Psychology of Popular Media Culture Video Game Players ' Personality Traits : An Exploratory Cluster Approach to Identifying Gaming Preferences Video Game Players ' Personality Traits : An Exploratory Cluster Approach to Identifying Gaming Preferences," *Psychology of Popular Media Culture*, vol. 9, no. 4, 2019.

[2] D. Y. Wohn, R. Ratan, and L. Cherchiglia, "Gender and genre differences in multiplayer gaming motivations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12211 LNCS, pp. 233–248, 2020, doi: 10.1007/978-3-030-50164-8_16.

[3] G. M. Troiano *et al.*, "Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development," *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–17, 2020, doi: 10.1145/3313831.3376755.

[4] B. Chrismanto, L. Toruan, and W. Istiono, "Enhancing Gaming Performance : A Recommender System for Selecting Optimal Gaming Headsets Based on SAW Method," *Journal of Advances in Mathematics and Computer Science*, vol. 38, no. 9, pp. 230–247, 2023, doi: 10.9734/JAMCS/2023/v38i91818.

[5] D. Kim, J. E. K. Nam, and C. Keum, "Adolescent Internet gaming addiction and personality characteristics by game genre," *PLoS ONE*, vol. 17, no. 2 February, pp. 1–14, 2022, doi: 10.1371/journal.pone.0263645.

[6] D. Kao, "The effects of juiciness in an action RPG," *Entertainment Computing*, vol. 34, no. February, p. 100359, 2020, doi: 10.1016/j.entcom.2020.100359.

[7] T. N. Luluquisin *et al.*, "Beast chasers: A 3D PC-based third person action RPG game used to spread societal issue awareness," *2021 11th International Workshop on Computer Science and Engineering, WCSE 2021*, no. Wcse, pp. 201–206, 2021, doi: 10.18178/wcse.2021.06.030.

[8] I. S. Baker, I. J. Turner, and Y. Kotera, "Role-play Games (RPGs) for Mental Health (Why Not?): Roll for Initiative," *International Journal of Mental Health and Addiction*, no. 0123456789, 2022, doi: 10.1007/s11469-022-00832-y.

[9] R. H. P. Prager, "Exploring the Use of Role-playing Games in Education," *Master of Teaching Research Journal*, vol. 2, no. 2, 2019.

[10] P. C. Joni, "Analisis Kualitas Software Pada Pembangunan Mobile Game RPG Berdasarkan Kebutuhan Kualitas Untuk Mobile Game," *It Journal Research and Development*, vol. 3, no. 1, pp. 62–71, 2018, doi: 10.25299/itjrd.2018.vol3(1).1901.

[11] F. Rozi and F. Firdausiah, "Implementation of Role-Playing Games in Overcoming Introverted Children," *AL-ISHLAH: Jurnal Pendidikan*, vol. 13, no. 2, pp. 1394–1402, 2021, doi: 10.35445/alishlah.v13i2.629.

[12] M. Grande-de-Prado, R. Baelo, S. García-Martín, and V. Abella-García, "Mapping role-playing games in Ibero-America: An educational review," *Sustainability (Switzerland)*, vol. 12, no. 16, 2020, doi: 10.3390/SU12166298.

[13] J. Sirota, V. Bulitko, M. R. G. Brown, and S. Poo Hernandez, "Towards procedurally

generated languages for non-playable characters in video games," *IEEE Conference on Computatonal Intelligence and Games, CIG*, vol. 2019-August, pp. 1–4, 2019, doi: 10.1109/CIG.2019.8848093.
[14] K. Fathoni, R. Y. Hakkun, and H. A. T. Nurhadi, "Finite State Machines for Building Believable Non-Playable Character in the Game of Khalid ibn Al-Walid," *Journal of Physics: Conference Series*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012018.
[15] G. A. da Silva and M. W. de Souza Ribeiro, "Development of Non-Player Character with Believable Behavior: a systematic literature review," pp. 319–323, 2022, doi: 10.5753/sbgames_estendido.2021.19660.
[16] O. R. Chandra and W. Istiono, "A-star Optimization with Heap-sort Algorithm on NPC Character," *Indian Journal Of Science And Technology*, vol. 15, no. 35, pp. 1722–1731, 2022, doi: 10.17485/ijst/v15i35.857.
[17] G. T. Kumala and W. Istiono, "Comparison of Flow Field and A-Star Algorithm for Pathfinding in Tower Defense Game," *International Journal of Multidisciplinary Research and Analysis*, vol. 5, no. 9, pp. 2445–2453, 2022, doi: 10.47191/ijmra/v5-i9-20.
[18] J. H. Kim, J. Lee, and S. J. Kim, "Navigating non-playable characters based on user trajectories with accumulation map and path similarity," *Symmetry*, vol. 12, no. 10, 2020, doi: 10.3390/SYM12101592.
[19] A. Novita *et al.*, "Longda Xiokan : Journal of Mandarin Learning and Teaching Development Of Learning Media Based on Car Racing Games In Mandarin Subjects For Class X Pangudi Luhur Don Bosko High School Semarang," *Journal of Mandarin Learning and Teaching*, vol. 1, no. 2, pp. 43–53, 2022.
[20] S. R. Lawande, G. Jasmine, J. Anbarasi, and L. I. Izhar, "A Systematic Review and Analysis of Intelligence-Based Pathfinding Algorithms in the Field of Video Games," *Applied Sciences (Switzerland)*, vol. 12, no. 11, 2022, doi: 10.3390/app12115499.
[21] Z. Zhang, J. Jiang, J. Wu, and X. Zhu, "Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm," *ISA Transactions*, vol. 134, pp. 42–57, 2023, doi: 10.1016/j.isatra.2022.07.032.
[22] Z. Zhang, J. Wu, J. Dai, and C. He, "A Novel Real-Time Penetration Path Planning Algorithm for Stealth UAV in 3D Complex Dynamic Environment," *IEEE Access*, vol. 8, pp. 122757–122771, 2020, doi: 10.1109/ACCESS.2020.3007496.
[23] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020, doi: 10.1109/LRA.2020.3026638.
[24] A. Andreychuk, K. Yakovlev, D. Atzmon, and R. Sternr, "Multi-agent pathfinding with continuous time," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-August, no. 1, pp. 39–45, 2019, doi: 10.24963/ijcai.2019/6.