

Maze Runner: Angel and Demon Path Finding Game Application using C-Theta* Algorithm

Silvester Dian Handy Permana¹, Ketut Bayu Yogha Bintoro², Budi Arifitama³, Ade Syahputra⁴
{handy@trilogi.ac.id¹, budiarif@trilogi.ac.id², ketutbayu@trilogi.ac.id³}

Informatics Study Program, Faculty of Creative Industry and Telematics, Universitas Trilogi¹²³⁴

Abstract. Pathfinding algorithm is an algorithm used to find the shortest path between two locations. This algorithm is used by some puzzle and strategy games. In maze game, pathfinding algorithm is used to find the shortest path to reach the finish point. However, the usual pathfinding algorithm has not been able to get the fastest path and computation so players need long waiting times in this game. In this research, the player becomes a "demon" that prevents the NPC from moving. The player is given several obstacles to be arranged so that the NPC takes the longest path of the possibilities. C-Theta* has the best ability to find the shortest path in the established grid. NPCs here become "angels" who are always looking for the shortest path from obstacles provided by players. This game asks for the longest time and distance from obstacles that can be composed by players.

Keywords: pathfinding, c theta* algorithm, game pathfinding, maze runner game.

1 Introduction

Path planning is a method of finding the optimal path from one point to another in pathfinding game. Path planning is necessary to optimize the path taken between the source and destination point. In addition, the path planning is done to see whether the NPC can move to the destination or not. This is because players in this game are given the opportunity to arrange obstacles so that the NPC can move with the furthest route. NPC is an autonomous robot because it utilizes a path search algorithm to determine it across points in the grid. Path planning can also be defined as the process of breaking the desired path movement into a number of repetitive steps to create a movement that optimizes the shortest path [1].

Barriers provided by players have an important role in road planning issues. The obstacle given to players are like the obstacles from Tetris the game. Tetris itself was a puzzle game designed and programmed by Alexey Pajitnov in June 1985, when he worked at the Dorodnicyn Computer Center at the Soviet Academy of Sciences in Moscow. Tetris has 7 blocks of formation in its game. The block resembles I, J, L, O, S, T, and Z. The block is called resembling because it is similar to the alphabet letter made using a small box / grid. At the start of the game the player will be given several blocks to be arranged freely and later the NPC will work to find the shortest path between the source point and the destination point.

However, the implementation of NPC planning concepts requires sophisticated modeling. Various path search algorithms such as Genetics, BFS, ANN and A* algorithms are used for road navigation on pathfinding problems. All of these algorithms assume the path between the origin and source point as a grid. The NPC later determines the path that has the

ICCSET 2018, October 25-26, Kudus, Indonesia

Copyright © 2018 EAI

DOI 10.4108/eai.24-10-2018.2280610

least number of grid cells to reach the destination [2]. Before C-Theta* algorithm, the A* algorithm was first created. The A* algorithm utilizes the heuristic function and calculates its value on each node in the work area to obtain the optimal solution but the A* algorithm has its limitations [3]. Thus the C-Theta* algorithm was developed as the development of the previous A* Algorithm. This study developed a computer algorithm that is a variant of the A* algorithm and that reduces processing time. The implementation of the C-Theta* algorithm gives players the experience to practice the ability to make the longest path of obstacles provided. This game makes the player a demon and NPC as an angel in angel and demon case problem. [4].

2 Literature Review

The research entitled C-Theta*: Cluster based Path-Planning on Grids was performed by [5]. This research has path planning to solve the problem in the form of transfer agent to the destination. The A*, Theta*, and C-Theta* algorithms are used to plan the angular paths that utilize lines during search. In the discussion, all the algorithms used are compared to get the longest and the fastest time on the way to the destination. In the results of his research, C-Theta* has a shorter path A* and the path is slightly longer than Theta*. In some other cases, the length of the C-Theta* is similar to Theta*. In the results for a 100×100 grid size of a 20% obstacle density, C-Theta* gets the same path length as Theta*. In maze the time taken by Theta* when compared with A* is more than doubled. When comparing Theta* with C-Theta* we see an increase of nearly 40% in the time taken and also the path shorter than A*. In more complex, detailed, and wider cases, C-Theta* is effective for finding shortest paths with the fastest time.

Research conducted by [6] in the title of Review of C-Theta* Algorithm and Prospects for Future Implementation reveals that C-Theta* is a new road planning algorithm with improved planning and planning speed compared to its predecessor, A* algorithm and Theta*. C-Theta* is the result of algorithm development of Theta* and Algorithm A* by utilizing Lazy-Theta*. Excellence in this method can find the shortest path and the fastest time than any other pathfinding algorithm. This research will be the basis for using C-Theta* in the formation of NPCs in order to find the fastest path.

Pathfinding in Games Research conducted by [7] reveals that artificial intelligence can be paired into games for commercial means. Particularly in the field of pathfinding, AI can help find the shortest path for the main characters in the game. The AI characters presented in this study are characters to match the players in finding the shortest path. This study proves that AI can be used in commercial facilities especially in the field of gaming. Academic and commercial research can be mutually beneficial to one another.

3 Research Methodology

The methodology constructed in this study is illustrated in Figure (1)



Fig. 1. Research Methodology.

Game built on Unity 2D Game Engine. Unity is a software used to create games. At the design stage is built first grid or block of maze with size 25 x 25 blocks. This size is the default size of the labyrinth. A 25 x 25 block can give freedom to a player who will put obstacles to block NPC from running. The next stage is the stage to create an obstacle or hurdle. Hurdles are built like Tetris blocks. Tetris is a computer game at the beginning of the game with the computer. Tetris will be built based on the letters I, J, L, O, S, T, and Z. This barrier is given by the player in blocking the search path of the NPC. The core part of this research is the application of the C-Theta* algorithm that the NPC will use to find its shortest path. In figure 4, is an example of the shortest path search using C Theta * between the two points. In figure 4 above it can be seen that the NPC will move from a green angle (beginning) to another green corner (end). The grid map is processed based on the size of the region. After the creation of the region, K-Means will be used to classify an area of high and low density based on the number of obstacles in a region. The purple stripe plotted from start to point is the path found by C-Theta*. To solve this path finding problem C-Theta* has been developed that tries to maintain the structural integrity of Theta* which makes it a popular path planning algorithm as well while trying to increase the time taken by Theta* to plan its course. For example in a 25 x 25 grid map it can be clustered 5 x5 to find the shortest possible path. Then there will be 125 small clusters to be computed. In this case, C-Theta* will use the k means to decide which region has the possible nearest point to one another. All data blocks will be stored in the array and the lowest density will be selected to find the path.

At the implementation stage, the researcher will change the Pseudocode Algorithm of C-Theta* in the selected programming language. In Unity, the compatible programming language is C#. Therefore, C-Theta* pseudocode above will be converted to C# programming language. After all the code is written, game elements such as buttons, scores, obstacles, and rules will be added. Once all the things are implemented then this game can be tested. The game will be tested by playing it and by observing the shortest path taken by the NPC. The further the path length of the NPC that will be taken then the more score will be earned by the players.

4 Discussion and Result

In this game, there are 2 parties involved namely NPC (Non Player Character) and the players. NPC is the angel character that will find the shortest path to reach its destination.

NPC is equipped with C-Theta* algorithm to find the shortest path. In this game, the player will act as the demon that has control over the place known as demon maze. An angel got lost inside the demon realm because of a certain event and as the demon, players need to interfere with the exit path. The demon has power to create a block-like obstacle. Players need to make it hard for the angel to find the exit by placing the obstacles in a certain way to block the path that will be taken by the angel and make it look for a longer route to the exit. However, the demon cannot completely block the path taken by the angel and make it impossible for the angel to reach the exit.

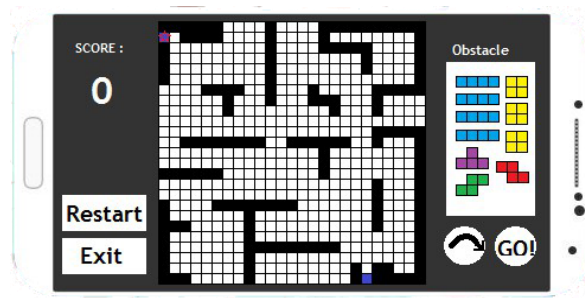


Fig. 2. The Game Interface.

Figure 2 above shows the gameplay interface. At the start of the game, the angel will be placed in a certain place in the map and it is shown as the blue colored square on the bottom-right side of the map. The players will then be given a set of obstacles that can be placed in the map by dragging the block to the maps. The obstacles will look like what the Figure 2 above shown on the right side of the game screen. Placed obstacles will then block the angel's path so that the angel have to find another path to reach the exit. The exit here is marked with the purple star on the top-left end of the map shown in Figure 2. Players have to cleverly put the obstacles in a certain way so that the angel take the longest path possible to the exit.

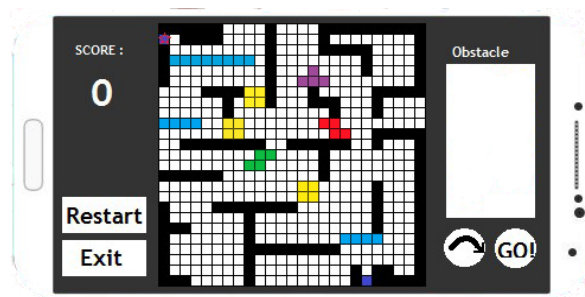


Fig. 3. Map with Placed Obstacles.

The game map will look like this after all of the obstacles are dragged and placed inside the map. Given the current map, the angel will then look for the shortest path to the exit. For every square passed the score will increment by 1.

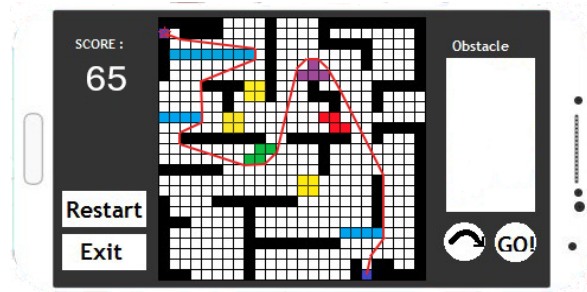


Fig. 4. Result of C-Theta* as Path Taken by Angel.

The red line drawn on Figure 4 above is the path taken by angel using the C-Theta* algorithm. By placing the obstacles like on the Figure 4 above gives a score of 65 to the player.

5 Conclusion and Suggestion

5.1 Conclusion

Based on the results of research, analysis, system design and programming until the completion stage of the program, the researchers can take the following conclusion:

- 1) C-Theta* provides the shortest path from the starting point to the destination point.
- 2) Pathfinding on game using the C-Theta* algorithm always manages to find the shortest path possible if there is a path from the starting point to the destination.
- 3) The times needed and the number of nodes examined for the search path depends on the distance between the starting point and the destination point.
- 4) The path resulted depends on the position of the starting point, destination point, obstacles and heuristic value.

5.2 Suggestion

The C Theta * algorithm can be modified with the Ant Colony Optimization Algorithm or Bee Colony Optimization Algorithm to find the best way if you have to go through the challenges. This modification will calculate each attribute of the challenge that will be faced so that's algorithm does not find the shortest path but the best way to go.

References

- [1] Z. Abd Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *Int. J. Comput. Games Technol.*, vol. 2015, 2015.
- [2] R. Graham, H. Mccabe, and S. Sheridan, "Pathfinding in Computer Games," *ITB J.*, vol. 4, no. 8, pp. 57–81, 2003.

- [3] S. D. Handy Permana, K. B. Yogha Bintoro, B. Arifitama, and A. Syahputra, "Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game," *IJISTECH (International J. Inf. Syst. Technol.*, vol. 1, no. 2, p. 1, 2018.
- [4] H. Reddy, "Path Finding - Dijkstra ' s and A * Algorithm ' s," pp. 1–15, 2013.
- [5] P. Mendonca and S. Goodwin, "C-theta*: Cluster based path-planning on grids," *Proc. - 2015 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2015*, pp. 605–608, 2016.
- [6] N. A. Phillip *et al.*, "Review Algoritma C-Theta * Dan Prospek Implementasinya Di Masa Depan," vol. 9, no. 2, pp. 129–137, 2017.
- [7] A. Botea, B. Bouzy, M. Buro, C. Bauckhage, and D. Nau, "Pathfinding in Games," *Artif. Comput. Intell. Games*, vol. 6, pp. 21–31, 2014.