International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Overview on Pathfinding Algorithm Between A* and Dijkstra's Algorithm for 2D Platformer Games

Pankaj Jakhar

Department of information technology B.K Birla college(Autonomous) Kalyan, Mumbai, India
mailto:pankajjakhar808@gmail.com

A B S T R A C T

AI has been a major research area in video games for decades. It is majorly used as the core of any Artificial Intelligence movement system in computer games. This paper compares the pathfinding algorithms employed in the platformer genre of video games and focuses on determining better-suited methods to be used. The previous researches show that A*(A-star) and Djikstra are the most suitable algorithms to be used for pathfinding. However, to make a decision between the two algorithms to be better suited to a particular environment can be difficult, thus, it raises further questions for which algorithm to be chosen for that particular environment.

Keywords:A* Algorithm, Dijkstra's Algorithm, Pathfinding, Platform Games

1. INTRODUCTION

One of the genres that exists within single player games is the platformers. Platformer games get their name from the fact that the game's character interacts with platforms [1]. In this type of games, players usually take control of one character and navigate them through a series of obstacles or puzzles that usually require jumping to overcome them. The most famous of this would probably be the Mario series of games in which the player takes control of the plumber Mario and navigates him through the game environment to reach the objective of the game. Despite graphics and technology have moved to mostly 3D games, this genre of 2D games still manages to find an audience.

The objective of this paper is to determine the best existing pathfinding algorithms that can be used in computer games for AI used for enemies in the game. This study will help beginner-level programmers or developers to understand what methods to use.

2. PATHFINDING ALGORITHMS

2.1 Dijkstra's Algorithm

Dijkstra's algorithm is used to find the shortest path from a starting node to a target node, by building a set of nodes that has a minimum distance from the source in a weighted graph. Dijkstra's algorithm can be found in [2]. Dijkstra is an uninformed/undirected search. The graph should have the followings:

- Either nodes or vertices that are then denoted with V or U in the algorithm.
- Two nodes that are connected by weighted edges in which (v,u) shows an edge and w(v,u) the height that denotes its weight.

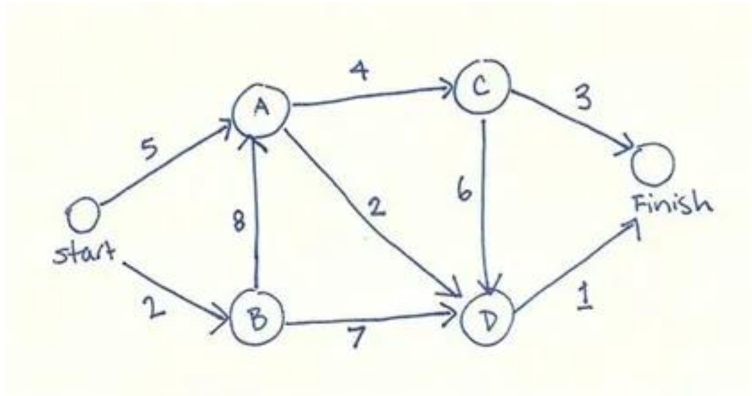

Figure 1 : Example of dijkstra's algorithm

Figure 1 shows the edges with the weights of each of them in grey writing. Essentially, the algorithm produces a shortest paths tree from the source, which is also known as a starting vertex or node, to every other node that is in the path. Published in 1959 and named after its Dutch computer scientist creator, Edsger Dijkstra, the algorithm can be applied on a weighted graph [2]. It is possible for the graph to be either undirected or directed. A condition to using this algorithm is that, on every edge, the graph is required to possess nonnegative weight. Figure 1 illustrates one such example of the graph with Dijkstra's algorithm.

An advantage for using Dijkstra's algorithm is that, due to it being an uninformed algorithm, it does not require knowledge of the target goal and is optimal for cases when the distance between the nodes and the target cannot be estimated. The algorithm is also useful when there are multiple target nodes since it picks the smallest cost with each step and will cover a large portion of the map. However, Dijkstra's algorithm also has a disadvantage in which it is an inferior algorithm compared to A* that will be covered further in this paper. In addition, due to it being an incomplete algorithm, it is possible for it to never find a solution.

2.2A* Search Algorithm

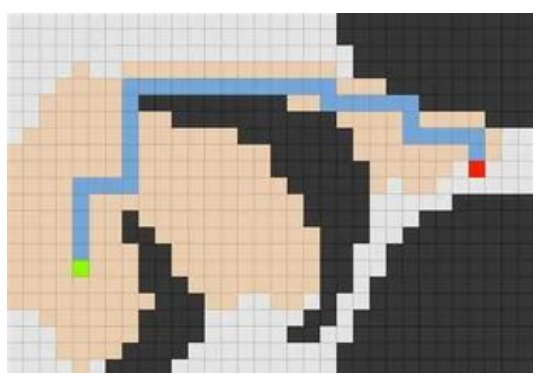

Figure 2 : A* algorithm in action

A* algorithm is an informed search algorithm that combines both features of Djiktra and Best First Search [3], meaning that it is formulated in terms of weighted graphs in which it picks the node at each step according to:
f = this is the sum of both g and h, which will be further discussed below.
With each step, it proceeds to pick the nodes/cells having the lowest f value and then processes those nodes/cells. Figure 2 shows an example of the A* algorithm in action.
'g' and 'h' are defined as follows:

● g = the cost of movement in order to move from the starting location to a specified location on the grid, which follows the path that was generated in order to reach the location.

● h = the cost of movement that is estimated in order to move from the specified location on the grid to the objective.

A* search algorithm is one of the more widely used algorithm techniques used in pathfinding and graph traversals due to its performance and accuracy. Peter Hart, Nils Nilsson, and Bertram Raphael of Stanford Research Institute (now SRI International) first published the algorithm [4].

Table 1 : Advantages and disadvantages of Algorithms

Algorithm	Advantage	Disadvantage
A*	• Will always find the solution and shortest path • Will only focus on the target node • Expands the fewest number of nodes to find the optimal path	• Complex to implement • Not optimal when there exists several target • Consumes a lot of memory
Dijkstra	• Will always find the shortest path • Picks edges with the smallest cost and is useful when there are multiple target nodes	• Blind search may consume time • It cannot handle negative edges • May fail if negative edges are involved

3. REVIEW FROM PREVIOUS RESEARCHES

As the research for the project was conducted, there were several past researches that are noteworthy to this topic, which may contribute to ideas or techniques that would be suitable for further research.

It is noted that the most common issue of pathfinding in a video game is how to avoid obstacles smartly and seek out the most efficient path over different terrains [5]. This means that the criteria to choose suitable algorithms is that it should be able to lead the NPC on the shortest path with the least computing time. The shortest path can mean both how fast the NPC navigates through the nodes and how many moves it needs to make in order to get the shortest path. However, there are other factors that may also need to be accounted for in order to determine the efficiency of an algorithm as shown by these past researches.

A. Paper Title: Comparative Analysis of Pathfinding Algorithms A*, Dijkstra, and BFS on Maze Runner Game
This paper by Silvester Dian Handy Permana, Ketut Bayu Yogha Bintoro, Budi Arifitama, and Ade Syahputra highlighted the Comparative Analysis of Pathfinding Algorithms A*, Dijkstra, and BFS on Maze Runner Game that is conducted to compare the efficiency of said algorithms on a 2D top-down maze runner game. In their experiments, he ran three different algorithms, which are A*, Dijkstra, and Breadth-First Search, and taking into account the computed blocks, the time taken to reach the goal, and the distance travelled by each algorithm.

Table 2: Example of a table from their research

Components	A* Algorithm	Djikstra's Algorithm	BFS Algorithm
Length	38	38	38
Time	0.3500ms	0.3000ms	0.8000ms
Computed Blocks	323 blocks	s 738 blocks	738 blocks

They made the conclusion that while all three algorithms can be used, A* is the best at pathfinding in maze and grid games and supported it with the minimal computing process needed by A* and the relatively short searching time compared to the other algorithms tested.

Acknowledgements

A SPECIAL GRATITUDE IS CONVEYED TO PROF. SWAPNA AUGUSTINE NIKALE, DEPARTMENT OF INFORMATION TECHNLOGY OF B.K BIRLA COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) KALYAN, THANE.

REFERENCES

[1] X. Cui and H. Shi, (2011)"A* - based Pathfinding in Modern Computer Games," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 11, no. 1

[2] Thaddeus Ably,Hannah Pang, Cristopher Williams,Jlmln Khlm,Eli Ross (2019) 'Dijkstra's Shortest Path Algorithm',brilliant.org

[3] Azad Noori ,Farzad Moradi (2015)'Simulation and Comparison of Efficency in Pathfinding algorithms in Games' DOI: 10.5902/2179460X20778

[4] Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics. 4 (2): 100–107.

[5] X. Cui and H. Shi, (2011)"A* - based Pathfinding in Modern Computer Games," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 11, no. 1