



Shortest Path Algorithm

PROJECT REPORT

Sir Sanjay | Discrete Mathematics | 12-05-22

Project Members:

1. Ammad Ali – 11096
2. Yumna Naz – 12414
3. Amit – 11259
4. Muhammad Nomir – 11330

Project Details:

In this project we will find out the shortest path between two nodes. We will create such an algorithm that will calculate the shortest path between two nodes and also show us that path. This algorithm will also be able to tackle extra nodes or obstacles in its path. We will be using the A* algorithm to calculate the shortest path which will achieve a better running time than Dijkstra's Algorithm. The selection of the nodes will be done on a graph so that user will not wander off the program.

In this project the user will select two nodes, Node A will be the starting point from where the path will start and Node B will be the point where the path will end. User will also be able to place obstacles to make the algorithm more interesting. Now when all of this is done then user will start the algorithm and the program will run which will then find out the shortest path and show the user.

Project Research:

Our current project stands on complete understanding of the A* algorithm so that we can use it properly in our project. Our group has developed the full understanding of A* algorithm and how does it work. We understood by taking reference from links below:

<https://www.educative.io/edpresso/what-is-the-a-star-algorithm>

<https://www.geeksforgeeks.org/a-search-algorithm/>

Our next and most important step was to implement this algorithm into our working project and develop a GUI for it so that it is more user friendly and not just a robust console-based project. For our GUI and code implementation we have taken the reference from the link below. Our GUI design will also be based on the below linked video, as it is the most close and easy representation of our project.

<https://www.youtube.com/watch?v=-L-WgKMFuhE>

To develop a GUI for our project we had to start from scratch because of the lack of information on how to develop a GUI we wanted, so after a fair amount of research we found a library of Python called pygame. It is a library which is used for visualization and small game developments and as we wanted to visualize our project to the user, we used this library which eventually led us to choosing Python for our project because previously we were intending to use C# but because of the lack of visualization options we chose Python. We found out about pygame by the link below:

<https://www.javatpoint.com/pygame>

<https://realpython.com/pygame-a-primer/>

We have also used the library of priority queue we have used this library because we want the value of 'f' to have the most priority while choosing the next node because the value of 'f' will always determine which node

to choose and if the 'f' of multiple nodes are same than 'h' gets the priority and in the end 'g'.

Our project in its final phase tells us the complete shortest path from one node to another and also tells us the unusable nodes within the completed path and alternative nodes that could have been used but weren't used because those nodes wouldn't have generated the shortest path and defeated the purpose of our algorithm and project. Every node has that performs a different purpose has been color coded so that it can be told apart.

Project Technique(s):

The main technique that we have used in our project is A* algorithm but to further break it down on how this algorithm works we are going to explain the technique in depth.

Let us take two nodes A and B. A is the start point where we want the algorithm to start and B is the end point, the destination where our algorithm will end. Now each node has 3 values, the first is 'g' it is the value that gives us a number which indicates the distance from start point to next node, the second is 'h' it is the value that gives us a number which indicates the distance from end point to the previous node disregarding the obstacles in between, the third is 'f' which is the value of 'g' and 'h' summed up and it is the value on the basis of which we select our next node, the smallest value would be our next node, although if there are many nodes with the same value of 'f' then we will consider the value smallest of 'h' and move to next node on that basis.

This is the basic technique we are and will be using in our project as this is the core of A* algorithm.

Project Software:

We are going to use the Python language and the IDE we are going to use is pycharm and on top of that we are going to use the library of pygame as it will help us in creating visualization/GUI for the user. We have also used the library of “Priority Queue” to give priority to different value of the nodes.

Project Relation With Course:

Our projects core relation with the course is to give a graphical representation of distance between nodes and finding the smallest path between them. Our project can be adjusted down to a simple technique of a graph having multiple nodes and each node have a cost to go to another node and this algorithm solves this graph by telling us the shortest path that can be taken in the graph to reach from one specific node to another. A more general example would be of maps that we use in our daily life it tells us the shortest path from our location to the destination.