International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | Jun 2022 www.irjet.net p-ISSN: 2395-0072

Node Path Visualizer Using Shortest Path Algorithms

Deep Singh[1], Brahmbind Singh[2], Gagandeep Singh[3], Harleen Kaur[4], Kanwarjeet Singh[5]

[1,2,3,4]B.Tech. I.T., Guru Tegh Bahadur Institute of Technology,GGSIPU New Delhi
[5] Assistant Professor, Department of Information Technology, Guru Tegh Bahadur Institute of Technology,GGSIPU,New Delhi

---

Abstract - *Visualizations of algorithms contribute to rising applied science education. The method of teaching and learning of algorithms is usually advanced and exhausting to known drawback. Visual image may be a helpful technique for learning in any applied science course. During this paper associate in nursing e-learning tool for shortest ways algorithms visual image is delineated. The developed e-learning tool permits making , reduction and saving graph structure and visualizes the rule steps execution. It is supposed to be used as a supplement to face-to-face instruction or as a complete application. The abstract pertinency of the delineated e-learning tool is illustrated by implementation of Dijkstra rule. The preliminary check results offer proof of economical mental models concerning shortest ways algorithms. This e-learning tool is meant to integrate completely different algorithms for shortest path.*

1. INTRODUCTION

Pathfinding or pathing is that the plotting, by a pc application, of the shortest route between 2 points. it's a a lot of sensible variants on determination mazes. This field of analysis is predicated heavily on Dijkstra's rule for locating the shortest path on a weighted graph. Pathfinding is closely associated with the shortest path drawback, at intervals graph theory, that examines the way to determine the trail that best meets some criteria (shortest, cheapest, fastest, etc.) between 2 points in a very giant network. At its core, a pathfinding technique searches a graph by beginning at one vertex and exploring adjacent nodes till the destination node is reached, typically with the intent of finding the most cost-effective route. though graph looking out strategies like a breadth-first search would realize a route if given enough time, alternative strategies, that "explore" the graph, would tend to achieve the destination sooner. Associate in Nursing analogy would be an individual walking across a room; instead of examining each potential route before, the person would typically enter the direction of the destination Associate in Nursing solely deviate from the trail to avoid an obstruction, and create deviations as minor as potential.

Algorithm: -

Edsger Dijkstra is Dutch. he's one amongst the large names in applied science. he's well-known for his handwriting and quotes such as

- Simplicity is requirement for reliable Ness.
- The question of whether or not machines will suppose is concerning as relevant because the question of whether or not submarines will swim.

1.1 Path Finding - Review

Methods as a lot of advanced issues arise or being developed in AI. An excellent deal of analysis work is finished in pathfinding for generating new rules that are quick and supply optimum path since the publication of Dijkstra algorithm in 1959. Most of the analysis work is valid exploration information. Therefore, the analysis should offer reliable and correct information as experiments are terribly volatile.

1.2 Representations

Pathfinding is employed in a very large choice as areas and typically enforced on completely different maps that are generated to check pathfinding algorithms. The wide widespread aps are enforce employing a grid-based graph, set of nodes and edges, represented within the rule. Usually, a grid is superimposed over a maps so the graph is employed to search out the optimum path. Most generally used represented are square tile grid which might either be accessed as-way path or 8-approach path. Each have their own blessings and drawbacks. Grid employed by researchers. Alternative are Maze patter, algorithm pattern, etc.

1.3 Need of Algorithms

For finding a path between 2 nodes in a very given graph a quest rule is needed.

Many searches rule is developed for graph-based pathfinding. Pathfinding rule typically finds the ways by increasing nodes and neighboring nodes in keeping with some given criteria. Pathfinding algorithm will be broadly speaking divided into Categories: up on and clothed pathfinding algorithms.

As the names suggested up on means that having previous information concerning the matter house before looking out it. Up on search refers to the utilization of information concerning the searched house like drawback map, calculable prices, associate in nursing estimate of goal location. Thus, the rule utilizes this information whereas looking out a path and it makes pathfinding quick, optimum and reduces memory usage in node enlargement. Numerous algorithms that be this class ae A*, Dijkstra's Greedy, Swarm and plenty of a lot. These algorithms use completely different heuristic operated or uniform price operate to utilize the functions.

2. LITERATURE REVIEW

Literature Review is needed to require the matter into concerns that can't be cleared within the past researches. several researchers attempt to interpret numerous quite conclusions and to boost those past results literature review is required. this literature serves several varied fascinating options, that forms the important background for the study and conducted a thought.

An important field of mathematical theory is that the mathematical study of the structure of abstract relationships between objects by means that of graphs (networks). though work of those constructions will be strictly theoretical, they'll be wont to model try wise relationships in several globe systems. one amongst most generally exploitation applications is determination of shortest ways in several sensible applications as: maps; automaton navigation; texture mapping; typesetting in TeX; urban traffic planning; optimum pipelining of VLSI chips; subroutines in advanced algorithms; telemarketer operator scheduling; routing of telecommunications messages; approximating piecewise linear functions; network routing protocols (OSPF, BGP, RIP); exploiting arbitrage opportunities in currency exchange; optimum truck routing through given hold up pattern.

2.1 DATA STRUCTURES

In practice, graphs are typically pictured by one amongst 2 commonplace information structures: closeness lists and closeness matrices. At a high level, each information structures are arrays indexed by vertices; this needs that every vertex encompasses a distinctive number symbol between one and V. in a very formal sense, these integers are the vertices.

2.2 ADJACENCY MATRICES

The other commonplace arrangement for graphs is that the closeness matrix, 1st projected by Georges Brunel in. The closeness matrix of a graph G may be a may be a V matrix of 0s and 1s, commonly pictured by a two-dimensional array A[1 .. V, 1 .. V], wherever every entry indicates whether or not a selected edge is gift in G. Specifically, for all vertices u and v if the graph is adrift, then A[u, v] := one if and as long as actinic radiation a pair of E, and if the graph is directed, then A[u, v] := one if and as long as actinic radiation a pair of E. For adrift graphs, the closeness matrix is usually bilateral, which means A[u, v] = A[v, u] for all vertices u and v, as a result of actinic radiation and vu are simply dierent names for an equivalent edge, and also the diagonal entries A[u, u] are all zeros. For directed graphs, the closeness matrix could or might not be bilateral, and also the diagonal entries could or might not be zero. Given Associate in Nursing closeness matrix, {we can/we will/we are able to} decide in ⇥ (1) time whether or not 2 vertices are connected by a footing simply by wanting within the acceptable find time for the matrix. we are able to additionally list all the neighbours of a vertex in ⇥ (V) time by scanning the corresponding row (or column). This period of time is perfect within the worst case, however though a vertex has few neighbours, we have a tendency to still got to scan the complete row to search out all. Similarly, closeness matrices need (V2) house, notwithstanding what number edges the graph truly has, in order that they are solely space-evident for terribly dense graphs.

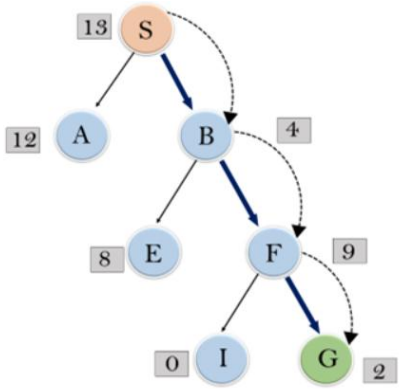


Fig.1 Dijkstra's working

3. Methodology

In this section, the operating of the project has been delineated, however the project started and the easy the project works and the way the varied phases of project were administrated and also the challenged round-faced at every level. What will the project do? At the essential, a pathfinding seeks to search out the shortest path between 2 points, this project visualizes numerous pathfinding algorithms in action and a lot of. The algorithm during this project has been custom-made and for 2nd grid, whenever ninety degree turns have a "value" of one and movements from a node to a different have a "value" of one.

3.1 Picking an Algorithm

Choose associate in nursing rule from the "Algorithms" menu. Note that some algorithms are unweighted, whereas others are weighted. Unweighted algorithms don't act or weighted node into consideration whereas weighted ones do. Also, not all algorithms guarantee the shortest.

3.2 Algorithms Description

Dijkstra's Algorithm: Dijkstra's rule permits US to search out the shortest path between any 2 vertices of a graph. It differs from the minimum spanning tree as a result of the shortest distance between 2 vertices won't embody all the vertices of the graph.

Greedy Best 1st Search: Greedy best-first search rule continually selects the trail that seems best at that moment. . it's the mixture of depth-first search and breadth-first search algorithms. It uses the heuristic operate and search.

Breadth 1st Search algorithm: an excellent algorithm; guarantees the most briefest way.

Depth first search algorithm: This is a algorithm for crossing or looking through tree or diagram. The algorithm begins at the root hub and searches quite far along each branch prior to backtracking. Doesn't ensure the most limited way.

3.3 Adding walls

Click on the grid to add a wall. Walls are unattackable and that implies that a way can't get through them. The mazes are presented and are visualized by choosing the mazes button on navbar and will be seen on the screen, clear dividers and loads, clear whole board, and change the imagine speed all from navbar.

3.4 Objectives of the project:

- It can be used as a E learning tool to understand Algorithms.
- It can used in finding Shortest Path.
- It can used in the telephone network.
- It can used in IP routing to find Open shortest Path First.
- It can used in geographical Maps to find locations of Map which refers to vertices of graph.
- We can make a GPS system which will guide you to the locations.
- Search engine crawlers are used BFS to build index. Starting from source page, it finds all links in it to get new pages.
- In peer-to-peer network like bit-torrent, BFS is used to find all neighbor nodes.
- As users of wireless technology, people demand high data rates beyond Gigabytes per second for Voice, Video and other applications.

3.5 Phases of the project

There are six stages presented for advancement of tasks. These are cooperated with all tasks, starting from information assignment and handling to the result for the user.

The phases are:

1. Building of Graph Matrix.
2. Added Walls and Event listeners.
3. Embed the Graph Algorithms.
4. Integrated the Path finding Functionality.
5. Improved the Design and UI.
6. Added the Timer Functionality.

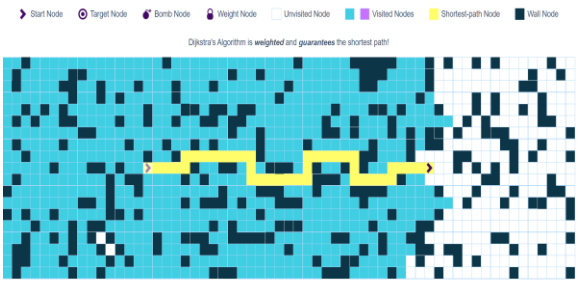
4. Results

Thus, we had the option the visualize all the way from a source node to the objective node effectively utilizing various kinds of calculations.

First snap on start and select a calculation.

Insert walls on the board grid 1

Fig.2: Insert walls at random grids.



Fig,3: Path finds by Dijkstra's algorithm

Fig.4: Representation of connected nodes

5. CONCLUSIONS

With the fruition of this project, we have effectively accomplished our target of our task is to implant Graph Path Finding with Visualization and Comparing their exhibition.

Just like with most other showing regions, there has been a critical hole between the hypothesis and functional comprehension of calculations acknowledgment. This is valid additionally for most brief ways calculations and specifically for Dijkstra calculation.

The principal objective of the task is to utilize it from activities research instructors and understudies for educating and concentrating on the current known combinatorial chart calculations.

The principal thought of the framework is to give a coordinated instructive climate to the two teachers and understudies to work with the educational experience in proficient manner.

To finish up, we have gained some significant experience of things working under this venture. We are likewise appreciative to our guide and manager for their endeavors in the educational experience.

REFERENCES

[1] https://www.geeksforgeeks.org/graph-data-stru cture-and-algorithms/

[2] https://en.wikipedia.org/wiki/Pathfinding

[3] https://www.w3schools.com/js/

[4] https://www.meta-chart.com/histogram

[5] https://www.geeksforgeeks.org/dijkstras-shorte st-path-algorithm-greedy-algo-7/

[6] https://youtube.com

[7] https://www.geeksforgeeks.org/a-search-algorithm/

[8] https://www.researchgate.net/publication/282488307_Pathfinding_Algorithm_Efficiency_Analysis_in_2D_Grid

[9] http://tesi.fabio.web.cs.unibo.it/twiki/pub/Tesi/DocumentiRitenutiUtili/p80-ardito.pdf

[10] Ahuja R. K., Magnanti, T. L. & Orlin, J. B. (1993). Network Flows: Theory, Algorithms and Applications. Englewood Cliffs, NJ: Prentice Hall.

[11] Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs. Numeriche Mathematik, 1, 269271.

[12] Fouh E., Akbar M. & Shaffer C. A. (2012). The Role of Visualization in Computer Science Education. Computers in the Schools, 29(1-2), 95-117.

[13] Roles J.A. & ElAarag H. (2013). A Smoothest Pathalgorithm and its visualizationtool. Southeastcon, In Proc. of IEEE, DOI: 10.1109/SECON.2013.6567453.