Path Explorer Visualization with Analysis

[1]Prof. Kanchan Umavane, [2]Mr. Dhanesh Sanjay Wadke, [3]Mr.Atharv Mohan Sawant, [4]Mr. Aniket Dilip Narangkar.

[1]Asst.Professor,[2'3'4]UG Student, [1'2'3'4] Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India.

[1]kanchanumavane2020@gmail.com, [2]dhaneshwadke59724@gmail.com,

[3]atharvsawant56459@gmail.com, [4]aniketnarangkar1@gmail.com

Abstract – Visualization is a technique which is used to learn a particular concept in competent way than conventional way . Contemporary advancements in technology have simplified tasks like creating learning models were once considered challenging. Visual aids that demonstrate how algorithms work contribute significantly to enhancing computer science education. Learning about algorithms can be challenging due to its complexity, but visualization is an effective technique for facilitating understanding in any computer science curriculum. This paper describes a tool that focuses on displaying various path-finding and sorting algorithms. Users can choose the algorithm and data, and interact with it to enhance their understanding [1]. The purpose of visualizer is to supplement teaching or to serve as a stand-alone approach for algorithm visualization. Understanding an algorithm by visualizing it with its time leads to a greater interest in understanding compared to traditional methods.

Keywords- Path finding algorithm, Dijkstra's , A*,BFS , DFS ,Visualization ,E-learning.

I. INTRODUCTION

Currently,there is a significant push towards promoting e-learning various fields. With modern technology, visualization tools can be developed to help explain topics such as various graph theory algorithms. Visual aids such as animations and illustrations play a significant role in increasing the effectiveness of the learning procedure. Such aids provide learners with more autonomy during the learning process. One example is algorithm visualization tools, which display the process of searching for the destination node by animating the traversal of nodes. When implemented effectively, these tools can bring algorithms to life and make difficult concepts more easily understood.[1] .Visual aids such as animations and illustrations play a significant role in increasing the effectiveness of the learning process. Such aids provide learners with more autonomy during the learning process. Comparitive analysis can be done for better knowledge of algorithms. One example is algorithm visualization tools, which display the process of searching for the destination node by animating the traversal of nodes. It can be used in GPS tracking system that is frequently used for obtaining live updates regarding a vehicle's location. This includes a GPS/GPRS module for acquiring location data and transmitting messages, as well as AT&T's cellular data service for transferring location information. The system provides accurate vehicle location information to the user based on data sent by the GPS Device.[4] When implemented effectively, these tools can bring algorithms to life and make difficult concepts more easily understood. Path finding visualizer is an approach to find the accurate path from start point to the end point using algorithms. A star is a popular algorithm to use, but from several research its seen that it has more potential in the path finding. So advancements have been made in the development of A-Star. To Find best algorithm Analysis will be done. Since the path will be same for all the algorithms the space complexity will be same. Different algorithm will take different time to reach from start node to end node, so analysis can be done based on time. Different speeds will be added to understand the working of algorithm properly.

II.AIMS AND OBJECTIVE

a)Aim

The aim is to obtain the desired outcome in the field of path finding. To understand and and learn effectively about different algorithms such as AStar's, Diijstra's , Bidirectional, Search, Depth-First Search known (DFS), Breadth-First Search (BFS).To compare an algorithm with other search algorithms and examine their potential future implications using time.

b) Objectives

The display shall be receptive to interaction through touch or pen input. Tailored to run smoothly on tablets and mobile with optimized efficiency. An explanation of both fundamental and complex path finding algorithms. It will make use of an API that makes it easier to check. Algorithm execution can be performed at different speeds, including fast, medium, slow, and step-wise execution. These options

DOI : 10.5281/zenodo.7863634

allow for better comprehension and understanding of the algorithm in question. The Grid system is customizable with options for Colours , Execution Speed, Pause, Resume and Stop.

III. LITERATURE SURVEY

Paper 1: Path E-learning Tool for Visualization of shortest path algorithms

It describes an e-learning Mechanism for visualizing shortest paths algorithms. It allows users to create, edit and save graph structures and shows the execution of algorithm steps. The tool is designed to be used either as a supplement to face-to-face instruction or as a stand-alone application. The usefulness of the tool is demonstrated by implementing the Dijkstra algorithm. Preliminary test results suggest that the e-learning mechanism is easy to use and has the potential to help students develop effective models of paths finding algorithms. This online educational resource aims to combine various methods for identifying the shortest route possible between points. [1]

Paper 2: The review and evaluation of shortest path algorithms

The aim of this research was to recognize and distinguish the concepts involved in the problem of finding the shortest path. Understanding the graph's basic terminologies and representation in the computer is crucial to solve the path searching problem. In addition to explaining the general concepts, this writing sheds light on the implementation of various algorithms such as Floyd-Warshall Algorithm, Dijkstra's Algorithm, Genetic Algorithm, and Bellman-Ford Algorithm. The performance evaluation of each algorithm is also presented in it.

Paper 3 : AlgoRhythm - A Sorting and Path-finding visualizer tool to improve algorithms teaching methodologies

This manuscript presents an overview of the design of AlgoRhythm, a tool that concentrates on different path-finding and sorting algorithms displayed on a canvas, providing users with the opportunity to modify code and data, and interact to enhance their understanding. The intended purpose of AlgoRhythm is to supplement teaching or to function as a standalone approach for the algorithm visualization. An investigation with it is depicted to indicate that the use of this tool has a higher interest in understanding than traditional methods.[2]

IV. EXISTING SYSTEM

In current system, algorithms used to plan path include several classification methods, such as graph search-based algorithms, vertex based, and heuristic algorithms i.e Depth-First Search (DFS) algorithm, Breadth-First Search (BFS)algorithm, Dijkstra algorithm and A* algorithm. Heuristic algorithms requires large amount of calculations, low efficiency, and poor collision-free performance[1]. The multiple algorithms are work on exiting system step-by-manner. For example , Dijkstra's Algorithm starts at source node and analyses the graph to find the shortest path between that node and all the other nodes in the graph. If it finds a shorter path, the node is marked as "visited" and added to the path. This process continues until all the nodes have been added, creating a path that connects the source node to all other nodes following the shortest path possible to reach each node.

V. COMPARATIVE STUDY

Sr. No	Paper Name	Author/ Publication	Technology	Advantage	Disadvantage
1.	Path Finding Visualizer Application for Shortest Path Algorithm	Nikhil Yadav, Karishma Dhameja; Prakhar Chaubey IEEE 2021	HTML, CSS, JavaScript	User-friendliness of the project provides user with easy instructions on how to operate it.	It does not Show time Required by Each algorithm to find the path from nodes.
2	Elearning Tool for Visualization of shortest path algorithms	Borissova,Mustakerov - Trends Journal of Sciences Research, 2015	HTML, CSS, JavaScript	Helps solving of implemen-ted combinatorial graph problems	Works only for simple astar algorithms and not for heuristics.
3.	The review and evaluation of shortest path algorithms	Kairanbay Magzhan,Hazar mat Jani ,2013	HTML,CSS, JavaScript	Helps to understand shortest path algorithms	Only provides theoretical knowledge and fails to visualize the algorithms
4.	AlgoRhythm - A Sorting and Path-finding visualizer tool to improve algorithms teaching methodologies	Arpit Trivedi; Kartikey Pandey; Vaibhav Gupta; Mukesh Kumar Jha, 2023	Html,css, javascript	Users can play around, interact, and modify both code and data to learn	No analysis is done for algorithms.

Table 1 : Comparative study

VI. PROBLEM STATEMENT

The existing system requires a large amount of calculations and a long calculation time. Thus A* and its variants are state-of-the-art algorithms for static environments to overcome Dijkstra's computational intensity. The A* algorithm can plan the shortest path in the map, but it needs

DOI : 10.5281/zenodo.7863634

to transit around the path nodes and select the minimum path cost which results in the efficiency of the algorithm will decrease with the expansion of the map scale. Thus, improved path planning algorithms are used to reduce algorithm processing time, increase smoothness, and shorten path length. In advanced system these algorithms also provide speed, time complexity, and knowledge about best shortest path algorithm to the individual[1].

VII. PROPOSED SYSTEM

The Path designing algorithms in this system use classification methods to differentiate based on environmental knowledge. These include graph search-based and classical algorithms such as Dijkstra's algorithm , DFS, bidirectional algorithm, BFS, and A* algorithm. To improve the existing system, improved path planning algorithms such as enhanced A* algorithm are used to minimize algorithm processing time and increase path smoothness, which is beneficial for mobile robot traversal in large workspaces. Also, space and time complexity is added to the system to measure the need for each algorithm[1].

VIII . ALGORITHM

Implementing Algorithm for Finding shortest path:

Step.1: Start

Step.2:

```
Declare all the required variable in constructor()
super("Dijkstra's Algorithm"):
    this.visitedNodes = null
    this.notVisitedNodes = null
    this.distance = null
    this.path = null
    this.parent = null
```

Step.3:

```
Initialize all the variable in firstFrame()
this.visitedNodes = new Set();
 this.notVisitedNodes=new;
Set([this.startNode]);
    this.distance = new Map();
    this.distance.set(this.startNode, 0);
    this.parent = new Map();
```

Step.4:

```
Write conditions for next iteration in perFrame().
const[min_dist,min_node]= this.getMinDistNode();
    this.visitedNodes.add(min_node);
    min_node? min_node.changeText(min_dist) : null;
    if (!min_node) {
      this.done();
      return;
    if (min_node == this.endNode) {
      this.done();
      this.mapPath();
      return;
 this.notVisitedNodes.delete(min_node);
  min_node.adjacents.forEach(n =>
this.notVisitedNodes.add(n));
 min_node != this.startNode && min_node != this.endNode
      ? min_node.setAsTraversed()
      : null;
this.notVisitedNodes.forEach(node =>
      const dist = this.distance.get(node) || Infinity;
      if (!this.visitedNodes.has(node) && dist > min_dist + 1)
        this.distance.set(node, min_dist + 1);
        this.parent.set(node.id, min_node);
If certain condition true call this.done()
And this.fixedFrame() will be called.
```

Step.5: run fixedFrame() for limited number after this.done()

Step.6: Stop

IX. MATHEMATICAL MODEL

Some of the algorithm used in advanced system are as follow:

1. A* algorithm :-

To speed up the search at the expense of optimality, ε-admissible algorithms are used. Weighted A*/Static Weighting's use

$hw(n) = \varepsilon\, ha(n), \varepsilon > 1$

as the heuristic function, and perform the A* search as usual. Dynamic Weighting uses the cost function

f(n)=g(n)+(1+\varepsilon w(n))h(n) and the depth of the search and N is the anticipated length of the solutions path. Sampled Dynamic Weighting uses sampling of nodes to better estimate.

2. Dijkstra's algorithm

Bounds of the running time of Dijkstra's algorithm on a graph with edges *E* and vertices *V* can be expressed as a function of the number of edges, denoted , and the number of vertices, denoted , using big-O notationIn the following, upper bounds can be simplified because is for any graph, but that simplification disregards the fact that in some problems, other upper bounds on may hold.For any data structure for the vertex set Q ,the running time is in

$\Theta(|E| \cdot T_{\mathrm{dk}} + |V| \cdot T_{\mathrm{em}}),$ When using binary heaps, the average case time complexity is lower than the worst-case.

X.SYSTEM ARCHITECTURE

In that ,Algorithms have a gap between theory and practical understanding. This is also true for shortest path algorithms and in particular for A* Algorithm. The main goal of the e-learning tool is to use it for studying known graph algorithms. Starting with the A*, other shortest path algorithms will also be implemented gradually like Dijkstra,bfs,dfs. The Pathfinding Visualizer tool involves three steps:

DOI : 10.5281/zenodo.7863634

l. Selecting Algorithm

This module provides the necessary capability to navigate through the nodes in a given structure.graph shortest path problem is solved by using grid and various techniques to solve the problem.It encompasses a range of algorithms, including Dijkstra's algorithm, A* and DFS.Algorithms are also involved in this matter. We can move in only four ways: Up, Down, Right, Left. This means that all the nodes are equally weighted.

2.Placing Nodes

When the algorithm initiates the search for the final destination,the traversal starts from the node specified in the function.This section of the course discusses the process of producing an animated presentation.It handles the animation process while plotting the path between points. The grid comprises of both source and destination nodes that are essential for the functioning of the system.

3.Visualizing

The PathVisualizer Component handles the mouse operations, implements the algorithms of algorithm module on the grid and handles all operation performed on the grid. Final path is found by visualizer using selected algorithm and the analysis will be given based on the time.

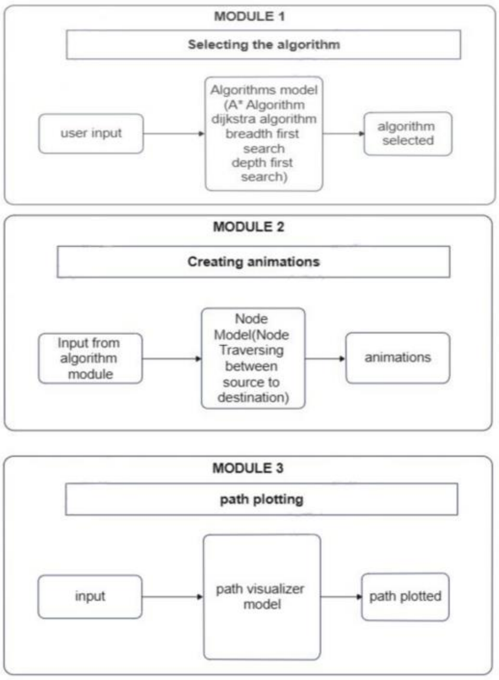

MODULE 1
Selecting the algorithm
user input
Algorithms model (A* Algorithm dijkstra algorithm breadth first search depth first search)
algorithm selected
MODULE 2
Creating animations
Input from algorithm module
Node Model(Node Traversing between source to destination)
animations
MODULE 3
path plotting
input
path visualizer model
path plotted

Fig 1 :- System architecture

XI.ADVANTAGES

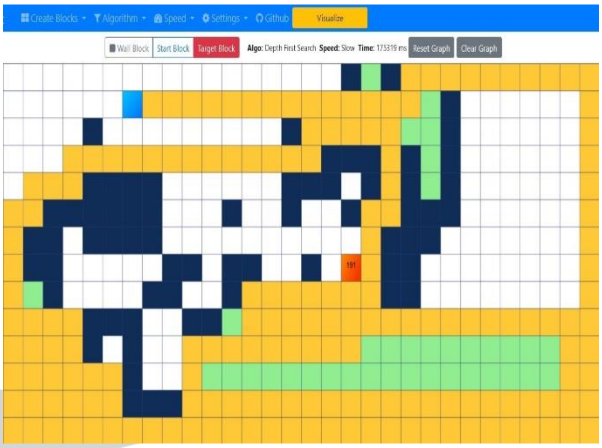
1.Visual aids are widely known as enhancement of learning capabilitie's. This tool simplifies the whole process of learning complex path finding algorithm and make it serviceable to rely on e-learning tools.

2.Visualization through different algorithms gives an individual more freedom in their learning journey and helps in finding more real life applications.

3.Visualization helps to construct knowledge and to organize the information. User having no knowledge about the topic can also learn by seeing.

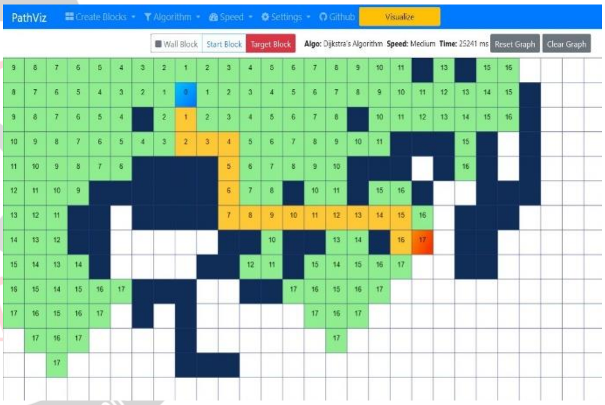
4.leads to better detention and also promotes active learning

XIl. DESIGN DETAILS



Algo: Depth First Search Speed: Slow Time: 175319 ms

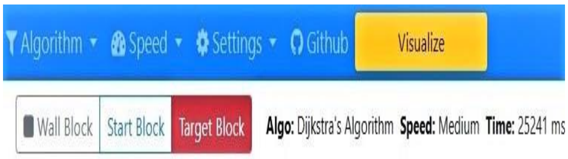
Fig 2 :- Result



Algo: Dijkstra's Algorithm Speed: Medium Time: 25241 ms

Fig 3 :- Result

The result shows visualization of different algorithms for finding path from start node to target node with their analysis by showing time in miliseconds.



Algo: Dijkstra's Algorithm Speed: Medium Time: 25241 ms

Fig 4 :- Result

XIII. CONCLUSION

Thus we ave tried to implement the paper “Path Finding Visualizer Application for Shortest Path Algorithm” , Nikhil Yadav, KarishmaDhameja; Prakhar Chaubey , IEEE 2021 & the conclusion is as follows This elearning is a modern solution of providing education and knowledge. Different tools are being made to implement this method of learning and make people realize its importance. e-Learmng

DOI : 10.5281/zenodo.7863634

tools accommodates everyone's need, from beginner to expert, and their consistency proves their effectiveness. This web application helps to visualize the working of pathfinding algorithms.Depth first search algorithm takes most time to visualize than other algorithm.further development in this project can include visualization of more complex algorithms, implementation over real world map.

REFERENCE

[1] Yadav, Nikhil, Karishma Dhameja, and Prakhar Chaubey. "Path finding visualizer application for shortest path algorithm." In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 1669-1672. IEEE, 2021.

[2] Trivedi, Arpit, Kartikey Pandey, Vaibhav Gupta, and Mukesh Kumar Jha. "AlgoRhythm-A Sorting and Path-finding visualizer tool to improve existing algorithm teaching methodologies." In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 158-169. IEEE, 2023.

[3] The review and evaluation of shortest path algorithms (https://www.researchgate.net/profile/MagzhanKairanbay/publication/310594546_A_Review_and_Evaluations_of_Shortest_Path_Algorithms/links/5ac49631a6fdcc1a5bd06106/A-Review-and-Evaluations-of-Shortest-Path-Algorithms.pdf).

[4] Prof. Vishal R. Shinde : “Real Time Bus Tracking System” in IJREAM, ISSN: 2454-9150 Special Issue-ICreate-2018, APR 2018. (indexed in Scope Database-Link:https://scopedatabase.com/documents/00000217/00000-99774.pdf)

[5] Steven HALIM, "VisuAlgo – Visualising Data Structures and Algorithms Through Animation", OLYMPIADS IN INFORMATICS, vol. 9, pp. 243-245, 2015.

[6] M. L. Cooper, C. A. Shaffer, S. H. Edwards and S. P. Ponce, "Open-source software and the algorithm visualization community", Sci Comput Program, vol. 88, pp. 82-91, Aug. 2014.

[7] D. Borissova and I. Mustakerov, E-learning Tool for Visualization of Shortest Paths Algorithms, 2015, [online] Available: http://www.tjsr.org

[8] Elearning Tool for Visualization of shortest path algorithms (https://www.tjsr.org/journal/index.php/tjsr/article/view/21)

DOI : 10.5281/zenodo.7863634