International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Data Visualization of Algorithms

Aman Keskar , Asmita Kulkarni, Jinendra Kulkarni
Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

Article Info
Volume 8, Issue 2
Page Number : 440-443

Publication Issue :
March-April-2022

Article History
Accepted: 10 April 2022
Published: 22 April 2022

ABSTRACT

Developing a Visualization System for Algorithms where different algorithms will be implemented considered and their functionality of how the shortest route is determined from the variety obstacles and representation of the shortest route.
The user will have the opportunity here to provide a flexible input for each algorithm to visualize.After receiving the start node,destination node and the obstacles the shortest path will be generated and with the help of animation will be displayed on the screen .The different algorithms have different modules for each of them. Algorithms -

- Dijkstra algorithm
- Greedy Best First Search Algorithm
- Breadth-First-Search Algorithm
- A * Algorithm
- Depth First Search Algorithm

This animation will introduce various algorithms used to determine the shortest route.
It will also show the various barriers that can be used to represent the shortest route. This project will help the user to understand the functionality of the various algorithms.

Keywords: Dijkstra's Algorithm, Greedy Best First Search Algorithm, Breadth-First-Search Algorithm, A * Algorithm, Depth First Search.

I. INTRODUCTION

Algorithm visualisations is a project that aims to improve computer science teaching. The process of teaching and learning algorithms is often a complicated and difficult to grasp subject. Visualization is a useful learning method in every computer science classroom. This An e-learning tool for visualising shortest route algorithms is described in this paper. The developed e-learning application allows for the construction, editing, and storage of graph structures, as well as the creation, editing, and storage of graph structures. The stages of the algorithm are visualised. It can be used on its own or as a supplement to other programmes. to one-on-one instruction The algorithms' implementation exemplifies the notion, applicability of the e-learning tool described.

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

II. METHODOLOGY

The model that we hav e used is the WaterFall model. Our project required us to complete each phase before the next phase can begin and there is no overlapping in the phases. So keeping that in mind we chose the WaterFall model.

The waterfall model depicts the software development process as a sequential flow of events. This indicates that any step of the development process can initiate only after the earlier one has ended. The phases in this waterfall model do not overlap. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase in sequence.

- Requirement Analysis: We gathered the requirements of the system to be developed and are captured in this phase.
- System Design: We studied the requirement specifications from first phase in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- Implementation: We will take the inputs from the system design, then system will be developed in small modules, which are integrated in the next phase. Each module is developed and will be tested for its functionality, which is referred to as Testing.
- Testing: All the units which will be developed in the implementation phase will be integrated into a system after testing of each module. Post integration the entire system will be tested for any faults and failures.
- Deployment: Once the functional and non-functional testing is done; then the product will be deployed in the customer environment or released into the market.
- Maintenance: After the deployment if any issue arises those will be addressed by the administrators. Also a monthly routine web-server maintenance will be done by us

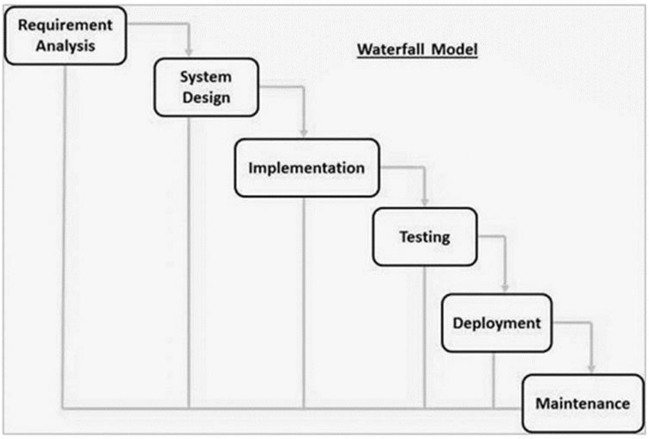


Fig 1. SDLC Model (Waterfall Model)

System Architecture:

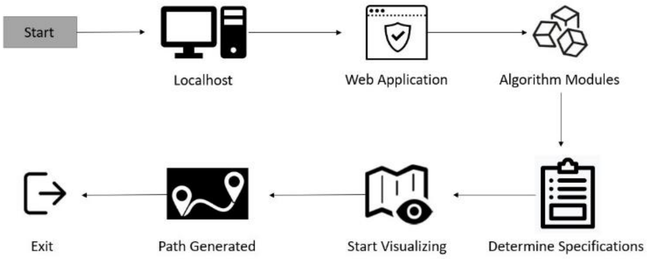


Fig 2. System Architecture Diagram

Dataflow :

- Local Host - Localhost is a hostname that refers to the current device used to access it. It's used to connect to the host's network services through the loopback network interface..
- Web Application - The project completely in the Web Development domain. It will be done using HTML,CSS, JavaScript.
- Choose Algorithm - There are various algorithms such as Dijkstra's Algorithm, BFS Algorithm, Greedy Algorithm, A* Algorithm. From the above given algorithms choose the one you want to visualize on the screen.
- Start Node - It's the starting point from where you want to find the shortest path.
- Destination Node - Its the end point of the path or also can be called as the point till where we want to find the shortest path.

- Obstacles Placement - After choosing the start node and the destination node we can place the obstacles in between and the shortest path is calculated considering all the obstacles chosen.
- Start Visualization - After the completion of all the above steps we can select Start Visualization and depending upon the Algorithm chosen it will start scanning the area in the grid.
- Path Generated - Path is generated and displayed on the screen.
- End - The process thus is completed.

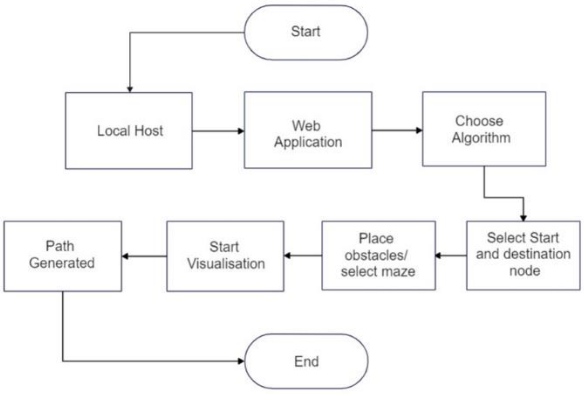


Fig 3. Dataflow Diagram

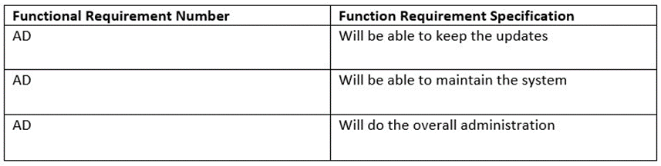
System Implementation:

- Task 1: To gather the requirements of the project.
- Task 2: To analyse the requirements.
- Task 3: To study and prepare the system design based on the requirements.
- Task 4: To implement the different algorithms.
- Task 5: To develop the system in modules.
- Task 6: To test the entire system for any faults and failures.
- Task 7: To deploy the entire system.
- Task 8: To do the web-server maintenance.

III. TABLES

Administrator Table:

The Administrator will be able to keep the updates and will to maintain the system and will do the overall administration



Functional Requirement Number	Function Requirement Specification
AD	Will be able to keep the updates
AD	Will be able to maintain the system
AD	Will do the overall administration

Fig 4. Administrator Table Diagram

End-user Table:

In our project we have 5 end-users and these end users have different functionalities

The functionalities of these end-users are as follows:

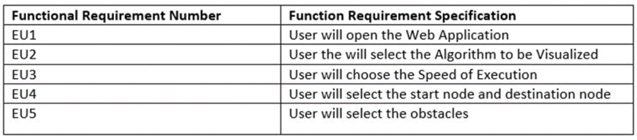
EU1 – The User will open the Web Application

EU2 – User will select the Algorithm to be Visualized

EU3 – User will choose the Speed of Execution

EU4 – User will select the start node and destination node

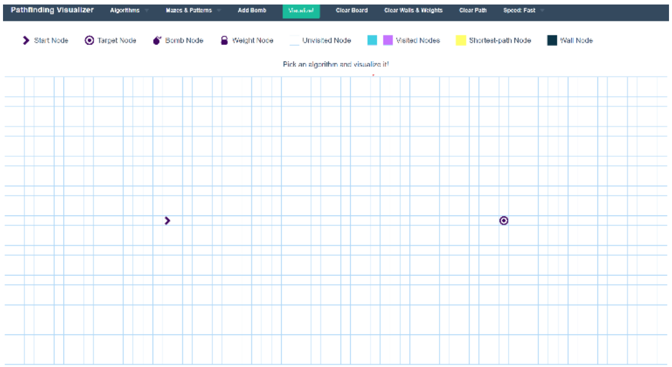
EU5 – User will select the obstacles



Functional Requirement Number	Function Requirement Specification
EU1	User will open the Web Application
EU2	User the will select the Algorithm to be Visualized
EU3	User will choose the Speed of Execution
EU4	User will select the start node and destination node
EU5	User will select the obstacles

Fig 4. End-user Table

IV. RESULTS



Pick an algorithm and visualize it!

Fig 5. Image of the Grid

This Image Describes the grid pattern of our project and the Start Node(Arrow Shape) and Destination Node(Target Shape).On the header there are different dropdown menus to select from.

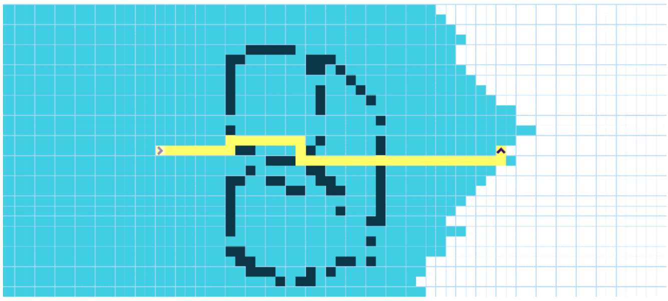


Fig 6. Image after Finding The Path(For Dijkstra's Algorithm)

The above image describes the scanning and the path after visualizing the Dijkstra's algorithm. The blue part on the grid is the scanned part. Once the destination node is scanned the path is generated from the start node to end node. The path here is in yellow colour. The Grey part is the obstacle in between the two nodes

V. CONCLUSION

- The described E-learning tool prototype is envisioned to support both of teachers and students.
- The implemented techniques for drawing over different graph layouts have been critically acclaimed by teachers and students.
- Different shortest path problems can be formulated and visualized.
- Since it's a e-learning tool it will be helpful for students and teachers in online education.

VI. REFERENCES

[1]. S. Diehl. Software View. Number 2169 in Lecture Notes in Computer Science. Springer, 2002.

[2]. E. Fouh, M. Akbar, and C.A. Shaffer. The importance of visualisation in computer science education. Computers in Schools, 29: 95–117, 2012.

[3]. S. Hall, E. Fouh, D. Breakiron, M. Elshehaly, and C.A. Shaffer. New education for data structures and algorithms studies. In the plans for the ASEE Annual Conference, Atlanta GA, June 2013.

[4]. C.D. Hundhausen, S.A. Douglas, and J.T. Stasko. Meta-analysis of algorithm recognition efficiency. Visual and Computer Languages Journal, 13: 259–290, June 2002.

[5]. V. Karavirta. Seamless integration of hypertext and animation algorithm. ACM switch In Computer Education, 9 (2): 1–18, 2009.

[6]. V. Karavirta. Location-based mobile reading of local algorithms. In IADIS forums 2013 International Telephone Learning Conference, pages 158–162, Lisbon, Portugal, March 2013.

[7]. V. Karavirta, A. Korhonen, and O. Sepp¨al¨a. Misconceptions in visual algorithm comparisons updated: UI impact on student performance, attitudes, and erroneous ideas Macau, 2013.

[8]. A. Korhonen. Visual Algorithm Simulation. Doctoral dissertation (tech rep. No. T4040), Helsinki University of Technology, 2003.

[9]. M. Krebs, T. Lauer, T. Ottmann, and S. Trahasch. View algorithm designed for students testing: flexible production, feedback and grading. In the 10th year scenes SIGCSE Conference on Innovation and Technology in Computer Science Education, pages 282-285, New York, NY, USA, 2005.

[10]. L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Sepp¨al¨a, and P. Silvasti. Visual algorithm simulates an exercise program with automation testing: TRAKLA2. Informatics Education, 3 (2): 267–288, September 2004.

[11]. T.L. Naps. Jhav'e: Support algorithm detection. IEEE Computer Graphics as well Applications, 25: 49–55, September 2005.