Web-based Visualization tool to demonstrate the working of pathfinding and sorting algorithms

Ankit Joshi
Dept. of information technology
Bharati Vidyapeeth College of Engineering, Kharghar
Navi Mumbai, India
e-mail:ankitjoshicr7@gmail.com

Nikhil Sawant
Dept. of information technology
Bharati Vidyapeeth College of Engineering, Kharghar
Navi Mumbai, India
e-mail: sawantnikhil122@gmail.com

Atharva Jagdale
Dept. of information technology
Bharati Vidyapeeth College of Engineering, Kharghar
Navi Mumbai, India
e-mail:atharvajagdale45@gmail.com

***Abstract*—This paper focuses on introducing a web-based visualization tool that demonstrates the working of famous sorting and pathfinding algorithms. These tools aim toward providing abstract thinking to different algorithms by making use of animations and 3D models which will stimulate the working of each separate algorithm in a unique way. To demonstrate the working of path-finding algorithms, a user-defined 3D playground will be made like a maze or a city where each algorithm will be animated in a unique way to show how it traverses to find the shortest path between two destinations. Similarly, each unique iteration of sorting algorithms will be animated by making use of 2D bars which will represent the array to be sorted. The animation tool would represent data as a bar graph and after selecting a data-ordering algorithm, the user can run an automated animation or step through it at their own pace. Each playground in the application will be user-defined to provide them with a powerful and interesting tool to learn and experiment with different algorithms. With the help of powerful frontend frameworks like React and WebGL Libraries like ThreeJs and React Fibre it will be optimized to work on low-end devices and mobile devices as well.**

Keywords—algorithm visualization, sorting algorithms, pathfinding algorithms, Vue.js, React.js, ThreeJS, WebGL.

I. Introduction

Data structures and Path Finding algorithms are of paramount importance in computer algorithms. Complete understanding of all these concepts is of the utmost importance for students which helps in developing problem-solving skills. It is important for IT Developers to solve problems with the help of programming languages, and data structures are key to solving these problems. Algorithms are the state of the art of computer science. There are so many ways of solving the same problem, and some approaches are better than others. Using the correct algorithm in the right place can save a lot of time as well as money.

For example, there are at least three or four ways to find the shortest path to reach the destination, but knowing that the A star algorithm is more efficient than the Dijkstra's algorithm, will lead to a much more efficient path-finding process, also it's fairly easy to implement a sequential or linear search algorithm for a list of data, but knowing that the binary search algorithm can sometimes be twice as efficient as the sequential search will lead to a better program.

Thus, by understanding the importance of data structures and pathfinding algorithms they have become an important part of computer science and technical recruitment. There are different learning methods like books, presentations, and online classes which are used by students, teachers, and developers to learn these algorithms. However, understanding the complete working of these algorithms requires conceptual thinking. To solve these problems recently there are various new 2D virtual tools emerging which explain the working of these algorithms step by step. However, understanding the working of a complex algorithm like graph traversals 2d faces some limitations. Taking advantage of advancements in WebGL we can upgrade these tools to 3D to give users in-depth working knowledge of these algorithms.

II. RESEARCH METHODS

There were three phases in conducting this research. In the first phase, a literature study was performed to derive a model based on existing related works and theories. The underlying important constructs and their relationships were required to be recognized first and understood well before they could be applied as a conceptual model to design the tool.

The second phase was designing the technologies that were to be used working on them to build the tool and discussing its scope. The identified constructs had to be supported by the tool, so later how they affected learning outcomes were possible to be examined. Therefore, the tool had to be designed and developed carefully so that could support the presence of all the identified constructs.

In the fourth phase, the tool was tested on learners to measure to what extent the tool, as the representation of cognitive support and engagement, influences learners' outcomes.

III. RELATED WORKS

Over the years, there have been many studies and papers on the use of sorting and pathfinding algorithms as visual aids. Some are comprehensive views on how to create animations and perform statistical analysis, and others focus on different techniques aimed for increased understanding of a similar animation. Several approaches have been conducted to bring algorithms in a more concrete form that can be easily understood by learners. Visualization is one approach that is commonly taken and programming, an activity that translates the algorithms into working programs, has been taught using the following approaches: traditional lectures and labs, robots, problem-based learning, software visualization, as well as programming environments and support tools. The teaching-learning process is still very laborious for teachers, from the preparation of good materials to the correction of large volumes of exercises among the publications involving support tools, 78.6% are related to the teaching of CS[3]. 3D implementation can be done in web application making the VR features available to most people[5].

The performance evaluation demonstrates that the method for on-demand loading of animation data improves the quality of the user interaction experience by reducing the amount of data loaded initially. However, the number of animation tracks and the size of the independent animation data affect the optimization of this method[1]. There is a method based on 3DS MAX technology to optimize the model memory. The modeling methods are based on the basic modeling library, composite modeling library, surface toads modeling library, polygon modeling library, facets modeling library, NURBS modeling library more than six types of model library and it identifies if the model is duplicated. Although, the research on model texture mapping technology still needs to be improved. Should focus on the optimization of texture mapping[12].

The focus of the interactive product lay in the process of product display, but also in real-time interaction on a system with the user also it is an important problem how to display the path for wase of understanding the path-finding algorithms [6]. Culling method and OpenGL ES API. Simplification of the scene by removing objects that are not currently visible or degrading the complexity of the models from the distance of the observer. Creating an image from 2D and 3D objects. Consequently, the computer graphics comes in with the employment of interface Open Graphics Library (OpenGL) for Android and new Metal API (application programming interface) introduced with IOS8. But the only issue here is level of detail (LOD) does not solve a problem that is directly visible in the graphic chain but it is perceptible in the complex scene[14].

The main advantage of rendered 3d maps is that some of the maps like Mapbox use a few simplifications for speeding up rendering. For example, when it is not possible to get the height information of a building, but it can be estimated to be 4 stories tall, developers can tag those width buildings: levels=4 in OpenStreetMap Mapbox Streets processing code will convert that to 12 meters in the vector tiles since it approximates the height of one level as 3 meters (about 10 feet)[2]. The optimization efficiency is not significant for a 3D model that contains only one animation track, and the animation data volume is small (e.g., less than 50 kb) if we consider using Java Applets, Swings, and AWT[4]. Direct Canvas mode has a great impact on performance and memory. Indeed, we can play WebGL games on mobile and the TV smoothly. Also, the amount of 3D memory decreased with Direct Canvas mode. That is because this mode does not create intermediate buffers with the window size anymore. But the only drawback is that there are already optimized WebGL renderers present like ThreeJs (a JavaScript library) which are specifically made for websites[7].

WebGL programs consist of control code is written in JavaScript and shader code that is executed on a computer's Graphics Processing Unit (GPU). As it is too complicated programming with WebGL directly, the framework three.js has been added to the system. Three.js allows the creation of GPU accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins[9]. The logical structure and physical storage format of the 3D spatial data, & can effectively divide and index the 3D scene. Results show that the proposed data structure can save about 60%-80% disk space, and the improved index structure is more sufficient[11].

IV. System Architecture

Effective instructional content will enhance learning outcomes. In a learning context, it is better not to use visualization solely without designing the instruction first since visualization is an instruction aid, not the instruction itself. For that, we have to keep the content updated regularly and easy to understand. To avoid confusion the conceptual model focuses on user interaction, developing and maintaining codebase, and visualization. The learner's engagements are through the blog posts, forums, and all the learning algorithms available in the tool/system. The user is any person exploring the tool. The Admin handles the backend i.e. codebase of the tool and updates it like adding a new algorithm or updating an existing one. The Database Manager's job is to keep the data of all users securely and safely.

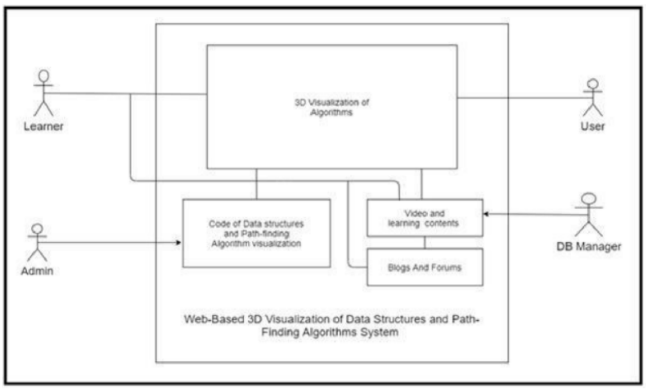
Learner
3D Visualization of Algorithms
User
Code of Data structures and Path-finding Algorithm visualization
Video and learning contents
Blogs And Forums
Admin
DB Manager
Web-Based 3D Visualization of Data Structures and Path-Finding Algorithms System

Fig. 1. A conceptual model of how algorithm visualization and learning instructions interact to improve learning outcomes.

V. Design Considerations

A. Implementation of the Sorting Algorithms

All the algorithms discussed are visualized in web applications by using bar charts. Each bar represents each element of the array. Once the animation starts, bars are swapped according to algorithm iteration thus visualizing the working of the algorithm. The animation speed of the tools can further be adjusted according to the user. Once the visualization of the algorithm finishes each iteration of the algorithm is also printed.

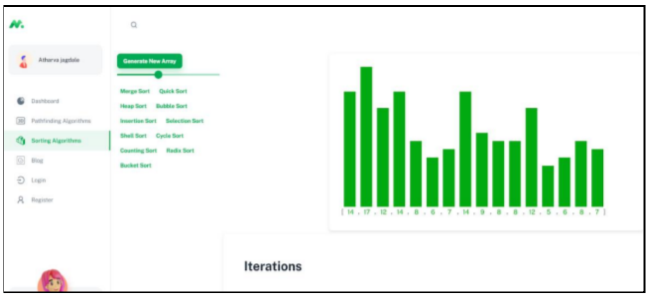

Fig. 2. Initial Random order.

Once the input and the sorting algorithm have been selected, the user can click the “Start” button in the row of buttons to see the sort run from beginning to end. To see the algorithm execution slowly step-by-step, the user can use slide-bar.

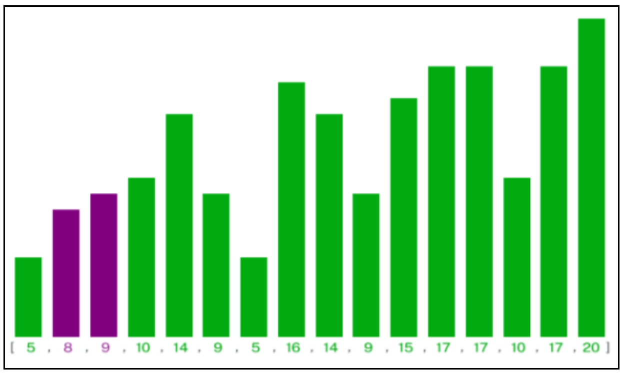

Fig. 3. During visualization.

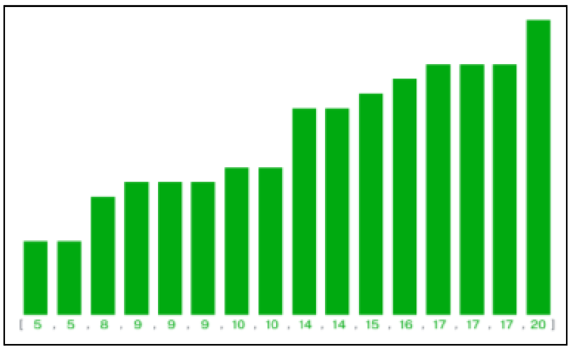

Fig. 4. Final State(After sorting).

This is significant to see how the movement of data can change depending on what algorithms had already affected it. It can also lend a different perspective on how the algorithms perform on semi-sorted data compared to the given ordering options.

B. *Implementation of the 3D grid using ThreeJS and react-threefiber*

Rather than using divs and spans, react-three-fiber (R3F) helps to render Three.js objects like meshes, lights, cameras, and shaders. We can declare the 3-D objects you want in a scene and R3F takes care of running the imperative Three.js code under the hood. And, in the same way, that React provides hooks to access the primitive DOM elements via refs, R3F lets you access the primitive Three.js objects for when you need a little more control.

In Order to visualize path-finding algorithms, we require a 3d grid that will represent different square nodes. This grid is made by two basic elements that are as follows: a. Geometry: It is a representation of mesh, line, or point geometry. Includes vertex positions, face indices, normal, colors, UVs, and custom attributes within buffers, reducing the cost of passing all this data to the GPU. Geometry for the grid is defined by the plane geometry class provided by ThreeJS. Parameters: width, height, width segments, height segment b. Material: Materials describe the appearance of objects. Properties: color, blending, etc

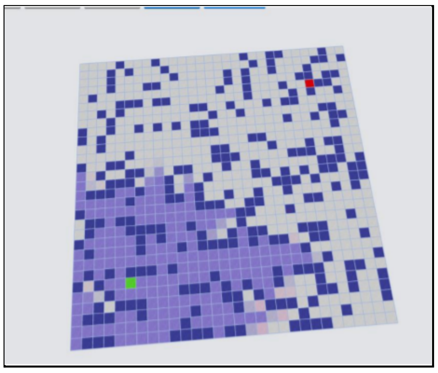

Fig. 5. During Visualization(animation)

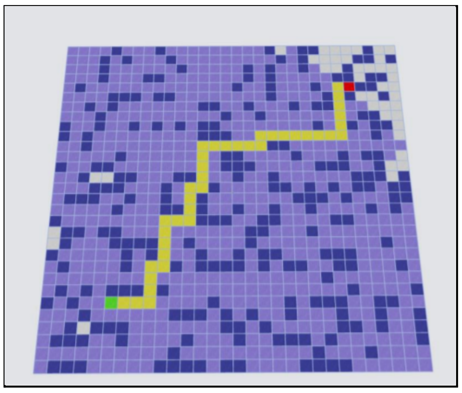

Fig. 6. Final State(reaching goal state)

a. **Random maze**: Random wall is generated using the Math, Random function. Each node is the grid can be accounted as a 2D array. This 2D array is iterated and is assigned with a wall or path. Here the blue boxes represent the wall. These blue walls are rendered using the TweenJS library.

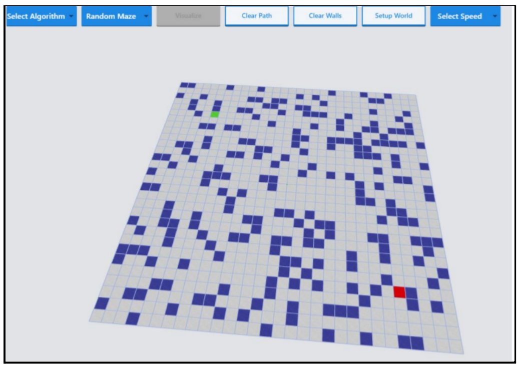

Fig. 7. Random Maze.

b. **Recursive maze**: The depth-first search algorithm of maze generation is frequently implemented using backtracking.
This can be described with the following recursive routine:
Algorithm:

```
1. Given a current cell as a parameter.
2. Mark the current cell as visited.
3. While the current cell has any unvisited neighbor cells.
   ● Choose one of the unvisited neighbors.
   ● Remove the wall between the current cell and the chosen cell.
   ● Invoke the routine recursively for a chosen cell which is invoked once for any initial cell in the area.
```

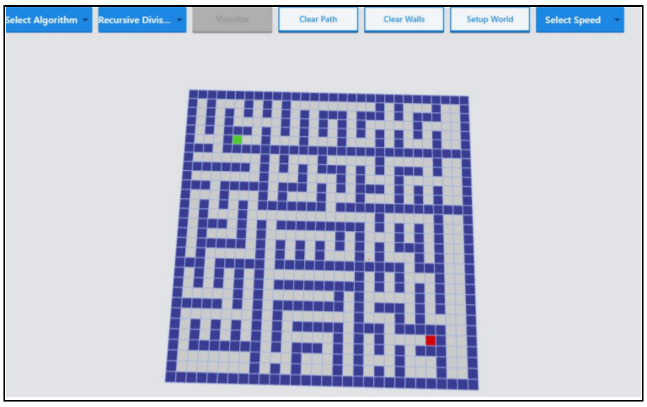

Fig. 8. Recursive maze

VI. Significance of the Tool

TABLE I. Table for time-taken by the tool to complete sorting visualization

No. of inputs	Algorithm visualization time (sec)					
	Bubble sort	Selection sort	Insertion-sort	Quick-sort	Merge-sort	Heap-sort
25	4.5	3.7	3	3.5	5	1.3
50	18.5	13.2	10	6.5	7.7	5.4
75	42.5	28.9	23.5	11	12.5	9.3
100	75	50	41	17	17.5	13.5

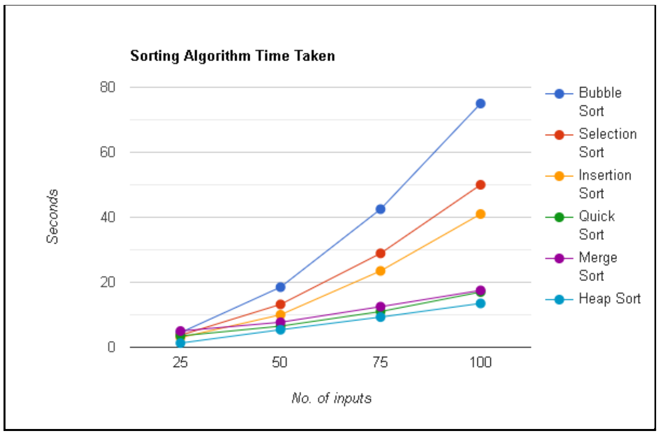

Fig. 9. Graph for time-taken for visualizing sorting algorithms.

TABLE II. Table for time -taken by the tool to complete pathfinding visualization

Maze Type	Algorithm visualization time (sec)			
	Dijkstar	A-star	BFS	DFS
Random Maze	22.4	18.3	24	68
Recursive Maze	27	26	32	76

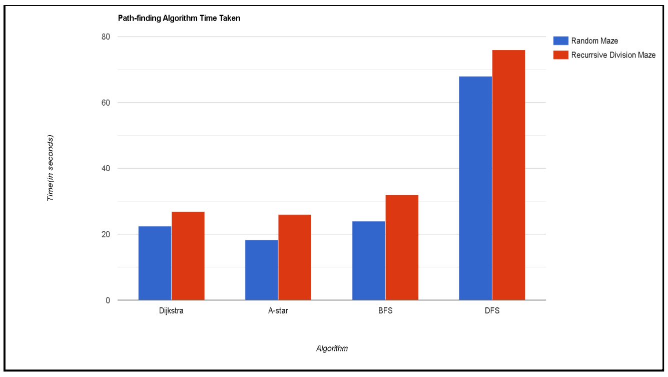

Fig. 10. Graph for time-taken to visualize pathfinding algorithm

Acknowledgment

We would like to thank our mentor Dr. S.M. Patil *Dept. of information technology Bharati Vidyapeeth College of Engineering, Kharghar* Navi Mumbai, India.

References

[1] L. Li, X. Qiao, Q. Lu, P. Ren and R. Lin, "Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading," in IEEE Access, vol. 8, pp. 88474-88486, 2020, doi: 10.1109/ACCESS.2020.2993613.

[2] G. Qiu and J. Chen, "Web-based 3D map visualization using WebGL," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018, pp. 759-763, doi: 10.1109/ICIEA.2018.8397815.

[3] D. B. Silva, R. d. L. Aguiar, D. S. Dvconlo and C. N. Silla, "Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students," 2019 IEEE Frontiers in Education

Conference (FIE), 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028702.

[4] Tao Chen and T. Sobh, "A tool for data structure visualization and user-defined algorithm animation," 31st Annual Frontiers in Education Conference. Impact on Engineering and Science Education. Conference Proceedings (Cat. No.01CH37193), 2001, pp. TID-2, doi: 10.1109/FIE.2001.963845.

[5] C. Peng, "The research and design Of 3D Web guide system based On WebGL," The 26th Chinese Control and Decision Conference (2014 CCDC), 2014, pp. 4052-4054, doi: 10.1109/CCDC.2014.6852890.

[6] Z. He, M. Shi and C. Li, "Research and application of path-finding algorithm based on unity 3D," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1-4, doi: 10.1109/ICIS.2016.7550934.

[7] H. Kim, S. Nam, J. Park and D. Ko, "Direct canvas: Optimized WebGL rendering model," 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1-3, doi: 10.1109/ICCE.2018.8326257.

[8] Á. Romhányi and S. Szénási, "OpenGL-based Modular Lightweight 3D Framework with Physics Capabilities," 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2019, pp. 239-244, doi: 10.1109/SAMI.2019.8782777.

[9] P. Li, X. Yu and J. Wang, "Progressive compression and transmission of 3D model with WebGL," 2016 International Conference on Audio, Language and Image Processing (ICALIP), 2016, pp. 170-173, doi: 10.1109/ICALIP.2016.7846665.

[10] W. Steingartner, J. Eged, D. Radaković and V. Novitzká, "Some innovations of teaching the course on Data structures and algorithms," 2019 IEEE 15th International Scientific Conference on Informatics, 2019, pp. 000389-000396, doi: 10.1109/Informatics47936.2019.9119320.

[11] J. Chen and W. Fang, "An Improved Spatial Data Structure for Web-based Large-scale 3D Scene," 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), 2019, pp. 681-684, doi: 10.1109/ICSESS47205.2019.9040768.

[12] H. Yutong, F. Wenlong and F. Yinshuang, "Research on model optimization based on 3DS max modeling," 2018 IEEE International Conference on Applied System Invention (ICASI), 2018, pp. 726-729, doi: 10.1109/ICASI.2018.8394362.

[13] H. S. Shin, H. I. Lee and E. S. Jang, "An effective data structure for a 3D printing slicer API," 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2016, pp. 1-4, doi: 10.1109/ICCE-Asia.2016.7804739.

[14] Marek T., Krejcar O. (2015) Optimization of 3D Rendering in Mobile Devices. In: Younas M., Awan I., Mecella M. (eds) Mobile Web and Intelligent Information Systems. MobiWIS 2015. Lecture Notes in Computer Science, vol 9228. Springer, Cham. https://doi.org/10.1007/978-3-319-23144-0_4

[15] Y. Liao, H. Matsui, O. Kreylos and L. H. Kellogg, "A Flow Visualization Using Parallel 3D Line Integral Convolution for Large Scale Unstructured Grid Data," 2018 IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV), 2018, pp. 106-107, doi: 10.1109/LDAV.2018.8739209.