Evaluating Software Algorithm Visualization Projects

Ashish Kumari[1], **Navdeep Bohra**[2],**Pardeep Sangwan**[3]

[1]*Department of IT, Maharaja Surajmal Institute of Technology, New Delhi, India*
[2]*Department of CSE, Maharaja Surajmal Institute of Technology, New Delhi, India*
[3]*Department of ECE, Maharaja Surajmal Institute of Technology, New Delhi, India*

Manuscript received May 01, 2023; Revised May 30, 2023; Accepted June 10, 2023.

Abstract: Algorithm visualizers (AVs) are very important tools to provide an understanding of how an algorithm works inside its core at each step. Therefore, we are at this moment demonstrating our findings and evaluating different algorithm visualizers that have been available in the computer education community for a time now. This paper is going to evaluate the algorithm visualizers that are publicly available. As the demand for digital technologies is increasing, better-skilled and performant engineers are required to be part of this digitization of the World. In pursuit of this, learning has to be very effective in computer science. To support this learning, too many algorithm visualizers are publicly available, fulfilling various requirements. After extensive search and evaluation of various algorithm visualizers, we found that most of the AVs available in the market need to be updated and need an upgradation like improvement in design, increase in user interaction, being platform independent, etc. This paper will reach its evaluation result using detailed parameters.

Index Terms: Algorithm, Visualization, Computer, Data structures, Evaluation

1. Introduction

Data structures and algorithms are as crucial for a computer science undergraduate student as programming languages. It helps create improved systems like FastTTps, etc. [1] . Learning these crucial topics is essential, but their dynamic nature with the data makes them very difficult to learn the implementation of data structures [2] . The abstractness of some algorithms makes it difficult for students to understand and remember the intricacy of the working of such algorithms [3]- [5] . Traditional methods like teaching code snippets from the book, writing code blocks with chalk on board or demonstrating the algorithms with limited and particular dataset were used by professors and educators. These methods don't let the naive understand the proper working of the algorithm with different types of datasets, the edge cases. This leads to a restricted understanding of the algorithms.

But, as the data structures can contain dynamic data and algorithms behave differently with different sets of data, some kinds of visual changes are required to improve the understanding of how an algorithm works at every step. There were many solutions found he researchers and educators to overcome this problem. Graphical representations, classical animations, and explanation videos were preferred techniques. But these techniques do not provide interactivity or diversity to work with these algorithms by varying their datasets as they are rigid presentations running on one single dataset [6] .

One of these great approaches is algorithm visualizers. The algorithm visualizes users with all the interactivity they need, animations, and the ability to tweak the base algorithm. This enables

the novice user to play with the dataset and the algorithm learning to understand better.

We have started our research with findings on different algorithm visualizers designed and programmed for public use. This Algorithm visualizer is an effective tool for understanding algorithms and data structures. These findings include algorithm visualizers based on technologies like Java, JavaScript, C, etc [9], [10].

Since starting, hundreds of Algorithm Visualizers have been created and designed by many of researchers and tech enthusiasts most of which needed to be recognized because of a lack of documentation and ownership. Thera are AlgoViz [7] , ANIMAL [8] , JAWAA [11] , Jeliot [12] , JHAVÉ [13] , TRAKLA2 [14] and JSAV [15] are the most popular and effective algorithm visualizers are available to the computing education community.

This paper focused on including the AVs with proper documentation and ownership accounts to increase reliability. It discussed ANIMAL [8] , JAWAA [11] , Jeliot [12] , and JHAVÉ [13] in comparative study.

Even after many years of use and effective learning from algorithm visualizers [16], there are active debates and arguments as to whether these visualization techniques are effective, or various studies have shown that there is “no difference” altogether concerning using visualization in algorithm learning [17] - [21] . While some studies have shown that visualizations in computer algorithms

To understand whether these algorithm visualizers are effective or not, we need to understand whether these algorithm visualizers follow some crucial requirements for an algorithm visualizer to be effective and useful. We picked some of these popular algorithm visualizers and evaluated them over predefined requirements. We are presenting our reviews and results over these evaluations in the subsequent sections of this paper.

2. Algorithm Visualizer

The authors should submit camera-ready paper in MS Word format by e-mail. Use A4 (210 x 297 mm) format of the paper. Leave 22,9 mm at the top, 18, 3 mm at the bottom, and 13,1 mm margins at left and right. The camera-ready paper size should be no more than eight pages without authors photos and CVs. Authors are strongly recommended to leave on the last page some place for photos and short author’s CVs. Please, don’t use numbering pages in your papers.

2.1. The history of algorithm visualization

Data structure and algorithm visualizers [16] have a long historical journey dating back to the 1940s [25]. But it was ahead of its time, as computer science was not much developed during those times and people were not much aware to this powerful concept of visualization. It was in 1984, that people understood the value of visualization in computer education.

The first algorithm visualization system was introduced in 1984, called BALSA (Brown Sedgewick, Brown University, USA) [26] . Since then, hundreds of algorithm visualizers have been developed and used, from small-scale supporting one or two algorithms to large-scale supporting numerous types of data structure and algorithms. Since then, Algorithm visualizers

have gained popularity, and extensive research was started to support this phenomenal cause of teaching. Some of the most popular and contemporary implementations or systems that made an impact were TRAKLA2 [14] , ANIMAL [8] , and JAWAA [11].

Early systems such as BALSA need to be more user friendly and their implementation of the technology is not suitable for the general public. In the mid-1990s when the internet and the World wide web started to become crucial parts of our lives, and java programming language is used to create platform independent software, visualization software also leaped their implementation and usability. As HTML5 and other web technologies gained popularity and web browser increased their capabilities of graphics rendering, the developers also started to build more sophisticated and performing applications for the web [24].

As these systems became readily available to more educators and students, their popularity of

them increased. A lot of commercial and non-commercial AVs were launched in the market. One of the reasons for the increase in popularity was that Java gave these systems [27] the feature of platform independence. A single developed application can run on different types of OS, making the user experience uniform.

2.2. Importance of algorithm visualizers

A computer science student must have knowledge of different algorithms and data structures. This knowledge is the most important part in order to develop effective and performing applications. With a sudden boom of jobs in the IT field, no of student learning programming is increasing rapidly. That creates a necessity of appropriate and effective methods for learning different algorithms in programming.

Most of the programming environments that are being used in institutions for learning are created for developing core software and applications [28] - [29] . They are not designed with the motive of educational purposes. This is considered the main barrier for most of programmers who are a beginner in programming. A much need better alternative is required to be specifically designed for educational purposes.

This is where the algorithm visualizer comes into the picture. It is a tool to represent various algorithms in graphical and animated formats for better learning for students. Numerous visualization systems [27] - [29] are developed and implemented for the computer science education community. The community has also praised this unique form of learning computer algorithms and data structures [30] . It has been proven to be more effective than using whiteboards or slides. According to various surveys, they can better attract students' attention during studying, explain concepts in visual and graphical terms, encourage a practical learning process, and helps in better communication between students and instructors. Interactive learning methods increase student concentration, and its visualization allows them to experiment and explore their concepts.

2.3. Work in field

There have been many ground-breaking visualization tools that have changed the course of the development of Avs [31] . But BALSA was the first to introduce the concept of algorithm visualization to the general public. It was a very ambitious project in the way that it commences the development of algorithm visualizers.

During the start of the internet era in the mid-1990s, a few he algorithm visualizer projects like JAWAA [11] and JHAVE [13] lead the way. Those were built on Java programming language and benefited from java-enabled browsers. Some key things that these visualizers needed to include interactivity and engagement in visualization. Instead, they focused particularly on creating a tool to make it easy for instructors to develop visualizations. They missed having an interactive and engaging software experience for the users. In mid 2000s, there were systems like TRAKLA2, which included specific exercises to complete for the participant. This made the system more engaging in learning [14] . It asks the user stop-and-think questions about the

algorithm or the data structure rendered on the screen. It also contains some tutorial text along with algorithm or data structure pseudocode to explain them. These are some good steps to engage the user of the application into the process of learning.

3. Study of algorithm visualizers

3.1. Selected projects

Although there are hundreds of algorithm visualizers of all types, scales and complexities, we picked major and ground-breaking projects from the last 20 to 30 years to make a comprehensive list of projects to be reviewed. Following are the names of the projects we selected to proceed further to our review. ANIMAL [8] , JAWAA [11] , Jeliot [12] , JHAVÉ [13]

3.1.1. ANIMAL [8]

Animal stands for “A New Interactive Modeler for Animations in Lectures”. It is written in Java programming language using the java swing library. Visualizers before this project were mostly required for educators or instructors to know the java language as they were built on java programming and only supported algorithm animation changes [22], [31] via some API calls. These calls have to be in java or the underline tech will not able to perform any task. The Animal offered a powerful set of graphical tools, by which animations can be generated and edited by using a drawing pane. Thus, it not needed any programming language by the instructor or user. ANIMAL supports inclusion of source code and pseudo code with highlighting. It also supports precise delays between actions. ANIMAL provides various more ways to create visualizations, as it has a graphical editor, customizable content generators and a scripting language for many algorithms.

3.1.2. JAWAA [11]

JAWAA is an easy-to-use, platform independent technology for creating visual animations of data structures and algorithms to run in web pages using java in web technology. The animations are implemented or written in a scripting language and could be easily generated as desired output from any other program written in another language. This lets the animator not need to know a java programming language to create animations in JAWAA. JAWAA needs one-liner commands to create and display a data structure or to do operations on that data structure, making the animations come alive. As JAWAA is written in a java programming language, it can be run on any operating system by using java’s platform-independent capability. Using java alongside Web technology makes creating animations in JAWAA easy and quick.

3.1.3. Jeliot [12]

Jeliot is a beginner friendly program animation tool. Its animations are basically execution of java programs in a step-by-step manner. Each step in the animation is graphically deducible and represents the interpretation of the program’s code. Although it was made to keep novice programmers in mind, it is shown in the studies that Jeliot is not that beginner friendly as it is announced [12] . Jeliot software animation tool is built on java programming language and hence supports platform independency, which means this software can run on any computer operating system. But, even coming this late in algorithm visualization, this tool needed to utilize the power of web application and java applet as others were doing at that time. This makes this application a little behind in terms of mass user reach and accessibility. One has to install java on their system to use this application.

3.1.4. JHAVÉ [13]

JHAVÉ stands for Java-Hosted Algorithm Visualization Environment, it is built with the sense of understanding that in the field of computer science algorithm education, only astonishing computer graphics, platform independency and ease of access are not enough for an algorithm visualization to succeed in teaching the learners about the subject. They understood that interactions with the tool, engagement with the visualization is just as important. JHAVÉ is built to support this ideology and it provides the engine to demonstrate graphical displays with inputs and in-between questions. It hooks the students in engaging more with the visualization and understand the process better.

3.2. Selected parameters

Selecting suitable candidates for our list of algorithm visualizers to evaluate and review is one part of the process. The other part is to define some requirements or parameters necessary for an algorithm visualizer to have or demonstrate. Since the start of this visualization software, they have been expected to demonstrate some key features. We are going to evaluate our pick of algorithm visualizers on the following requirements.

– R1. Platform Independent - An Algorithm visualizer should be platform dependent and easy to access.

- R2. Ease of Installation – Just like platform independency, ease of installing a soft‘ ware is also very important. User should be burdened to install lots of library and base images for the system to work.
- R3. Beginner Friendly – The system should be beginner friendly and easy to use in its educating factor.
- R4. Interactive Animation – Animation in the system should be interactive to be effective and engaging.
- R5. Questionnaires – Interactive questionnaires asked from time to time to the user can help them understand the concept better.

4. Result of evaluation

We have evaluated and tested these softwares in order to generate an evaluation table for displaying how well these softwares stand on our list of requirements. After evaluating these softwares on strictly on our requirements list, we found each of these visualizers are mising out on one or more important features.

We compiled the result into a table which contains all the requirements and visualizer software as rows and column respectively. In our findings, we can see that a lot of these software is not actively developed by their developers. Either these software were built on deprecated technologies or they are just left out for other projects. Recently Oracle Corporation has declared that java applets are deprecated now [33]- [35] .

We can also see that all of these software using java or web-based technology to support platform independency. This increases the reach of software visualization tools to the widest possible audience and benefits more learners. Some of these are also trying to incorporate the ease of installation to avoid any hassle for a novice computer learner the software before using it. It makes it easy for learners to try these applications with less efforts and focuses more on using them. It was found that only some software is needed to know some programming, scripting or command language to use them [6] . This can cause problems with novice learners as they could be not knowing any programming language with enough expertise to write the visualization from themselves [32] . Even after removing programming barrier from the software, only ANIMAL is a beginner friendly visualization tool.

Only JHAVE is the software that asks stop-and-think questions to the users in between the animations to make the learning more interactive. The animation itself is interactive in lot of these software, which makes the animation more engaging and give some level of controls to the learner also.

		ANIMAL	**JAWAA**	**Jeliot**	**JHAVÉ**
Platform Independent	R1	✓	✓	✓	✓
Ease of Installation	R2	✗	✓	✗	✓
Beginner Friendly	R3	✓	✗	✗	✗
Interactive Animation	R4	✓	✗	✓	✗
Questionnaires	R5	✗	✗	✗	✓

5. Conclusion

The field of computer science education is much more developed and improved from early 1990s when the first algorithm visualizer was launched. The Various number of software visualization tools with different approaches and technologies are developed from that time. But, very few of them are good enough to make an impact in data structure and algorithm learning for users. These were the groundbreaking projects with unique features and first-in-the-market ideas. Some of them are not in active development now due that have yet to be deprecated technologies from that time, making it difficult to upgrade them to modern technology. We found that as the tech of the industry’s improved, software visualization techniques also benefited from it and adapted the

technology from improvement their own.

References

[1] P. Agrawal, A. Zabrovskiy, A. Ilangovan, C. Timmerer and R. Prodan, FastTTPS: fast approach for video transcending time prediction and scheduling for HTTP adaptive streaming videos, Cluster Computing 24(3) (2021), 1605-1621.

[2] H. Danielsiek, W. Paul and J. Vahrenhold, Detecting and understanding students' misconceptions related to algorithms and data structures, 43rd ACM technical symposium on Computer Science Education (2012), 21-26.

[3] J. Talekar, J. Suthar, S. Joshi and J. Patel, Review of Algorithm Visualization Methodologies, International Research Journal of Modernization in Engineering Technology and Science, volume 4, issue 4 (2022).

[4] Chui, K. T., Kochhar, T. S., Chhabra, A., Singh, S. K., Singh, D., Peraković, D., ... & Arya, V. (2022). Traffic accident prevention in low visibility conditions using vanets cloud environment. International Journal of Cloud Applications and Computing, 12(1), 1-21.

[5] Riyahi, M., Rafsanjani, M. K., Gupta, B. B., & Alhalabi, W. (2022). Multiobjective whale optimization algorithm-based feature selection for intelligent systems. International Journal of Intelligent Systems, 37(11), 9037-9054.

[6] Kumar, S., Singh, S. K., Aggarwal, N., Gupta, B. B., Alhalabi, W., Band, S. S. (2022). An efficient hardware supported and parallelization architecture for intelligent systems to overcome speculative overheads. International Journal of Intelligent Systems, 37(12), 11764-11790.

[7] A. Kumari, M. Mittal, V. Jha, A. Sahu, M. Kumar, N. Sangwan and N. Bohra, Algorithm Visualization - Modern Web-Based Visualization of Sorting and Searching Algorithms, Advances and Applications in Mathematical Sciences, Volume 21, Issue 5 (March 2022), 2721-2736.

[8] G. Rößling, M. Schüler and B. Freisleben, The ANIMAL algorithm animation tool, 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education (2000), 37-40.

[9] Guida, C. G., et al. (2021, December). An Integrated BIM-IoT approach to support energy monitoring. In International Conference on Smart Systems and Advanced Computing (Syscom-2021).

[10] Dahiya, A., et al. (2021). A reputation score policy and Bayesian game theory based incentivized mechanism for DDoS attacks mitigation and cyber defense. Future Generation Computer Systems, 117, 193-204.

[11] W. C. Pierson and S. H. Rodger, Web-based animation of data structures using JAWAA, ACM SIGCSE Bulletin 30(1) (1998), 267-271.

[12] Moreno and M. S. Joy, Jeliot in a demanding educational setting, Electronic Notes in Theoretical Computer Science 178 (2007), 51-59.

[13] T. L. Naps, Jhavé: Supporting algorithm visualization, IEEE Computer Graphics and Applications 25(5) (2005), 49-55.

[14] Korhonen, L. Malmi, P. Silvasti, J. Nikander, P. Tenhunen, P. Mård, H. Salonen and V. Karavirta, Trakla2, 9th Koli Calling International Conference on Computing Education Research (2003), 43-46.

[15] V. Karavirta and C. A. Shaffer, Creating engaging online learning material with the JSAV javascript algorithm visualization library, IEEE Transactions on Learning Technologies 9(2) (2015), 171-183.

[16] Singh, I., Singh, S. K., Singh, R., Kumar, S. (2022, May). Efficient loop unrolling factor prediction algorithm using machine learning models. In 2022 3rd International Conference for Emerging Technology (INCET) (pp. 1-8). IEEE.

[17] J. S. Gurka and W. Citrin. Testing effectiveness of algorithm animation. In Proceedings, IEEE Symposium on Visual Languages, pages 182–189, 1996.

[18] C. Hundhausen and S. Douglas. Using visualizations to learn algorithms: should students construct their own, or view an expert's? In Proceedings, IEEE Symposium on Visual Languages, pages 21–28, 2000.

[19] 1. Gaurav, A., Arya, V., & Santaniello, D. (2022). Analysis of machine learning based ddos attack detection techniques in software defined network. Cyber Security Insights Magazine (CSIM), 1(1), 1-6.

[20] Peñalvo, F. J. G., Maan, T., Singh, S. K., Kumar, S., Arya, V., Chui, K. T., & Singh, G. P. (2022). Sustainable Stock Market Prediction Framework Using Machine Learning Models. International Journal of Software Science and Computational Intelligenc, 14(1), 1-15.

[21] D. J. Jarc, M. B. Feldman, and R. S. Heller. Assessing the benefits of interactive prediction using web-based algorithm animation courseware. In SIGCSE '00: Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education, pages 377–381, New York, NY, USA, 2000. ACM Press.

[22] Kaur, P., Singh, S. K., Singh, I., Kumar, S. (2021, December). Exploring Convolutional Neural Network in Computer Vision-based Image Classification. In International Conference on Smart Systems and Advanced Computing (Syscom-2021).

[23] S. R. Hansen, N. H. Narayanan, and D. Schrimpsher. Helping learners visualize and comprehend algorithms. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 2, 2000.

[24] P. J. Guo, Online python tutor: embeddable web-based program visualization for CS' education, 44th ACM Technical Symposium on Computer Science Education (2013), 579-584.

[25] S. Diehl (Ed.), Software Visualization, Lecture Notes in Computer Science 2269, 2002

[26] M.H. Brown, R. Sedgewick, A system for algorithm animation, Proceedings of the 11th annual conference on Computer graphics and interactive techniques, SIGGRAPH'84 (ACM New York, NY, USA, 1984)

[27] Singh, K., Setia, H., Kumar, S. (2021, December). Wi-Vi and Li-Fi based framework for Human Identification and Vital Signs Detection through Walls. In International Conference on Smart Systems and Advanced Computing (Syscom-2021).

[28] Sharma, A., Singh, S. K., Kumar, S., Chhabra, A., Gupta, S. (2023, February). Security of Android Banking Mobile Apps: Challenges and Opportunities. In International Conference on Cyber Security, Privacy and Networking (ICSPN 2022) (pp. 406-416). Cham: Springer International Publishing.

[29] Mengi, G., Singh, S. K., Kumar, S., Mahto, D., Sharma, A. (2023, February). Automated Machine Learning (AutoML): The Future of Computational Intelligence. In International Conference on Cyber Security, Privacy and Networking (ICSPN 2022) (pp. 309-317). Cham: Springer International Publishing.
[30] E. Vrachnos and A. Jimoyiannis, Dave: A dynamic algorithm visualization environment for novice learners, 2008 Eighth IEEE International Conference on Advanced Learning Technologies (2008), 319-323.
[31] Singh, S. K., Sharma, S. K., Singla, D., Gill, S. S. (2022). Evolving Requirements and Application of SDN and IoT in the Context of Industry 4.0, Blockchain and Artificial Intelligence. Software Defined Networks: Architecture and Applications, 427-496.
[32] Singh, A., Singh, S. K., Mittal, A. (2022). A Review on Dataset Acquisition Techniques in Gesture Recognition from Indian Sign Language. Advances in Data Computing, Communication and Security: Proceedings of I3CS2021, 305-313.
[33] Dalibor Topic, White paper Migrating from Java Applets to plugin-free Java technologies, Oracle Corporation, (2016).
[34] Al-Sharif, Z. A., et al. (2020). Live forensics of software attacks on cyber–physical systems. Future Generation Computer Systems, 108, 1217-1229.
[35] Almomani, A., et al. (2013). Phishing dynamic evolving neural fuzzy framework for online detection zero-day phishing email. arXiv preprint arXiv:1302.0629.