AAnim: An Animation Engine for Visualizing Algorithms and Data Structures for Educators

Zhuozhuo Joy Liu
zhuozhuocs@gmail.com
San Francisco State University
San Francisco, CA, USA

Timothy Sun
timothysun@sfsu.edu
San Francisco State University
San Francisco, CA, USA

ABSTRACT

Video streaming platforms are a major source of information for computer science students, where data structures and algorithms can be illustrated by means of animation. Our Python library, *AAnim*, aids a content creator in making such videos without any special expertise. Given a list of data structure queries or an input graph, our library creates a high-quality video that illustrates both how the data structure evolves and how these changes occur in the algorithm's pseudocode. The data structure's layout is generated automatically, further alleviating the difficulty in creating such videos. Example animations are available at https://youtube.com/playlist?list=PL-UJL8NI-eS5HQDoomg1rMfou5eO-OwuP.

CCS CONCEPTS

• **Applied computing → Computer-assisted instruction**.

KEYWORDS

algorithm visualization, data structure visualization, visualization, animation, computer science education

ACM Reference Format:
Zhuozhuo Joy Liu and Timothy Sun. 2023. *AAnim*: An Animation Engine for Visualizing Algorithms and Data Structures for Educators. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3545947.3576233

1 INTRODUCTION

Online learning has become more popular among computer science students, and in this realm there has been an increasing demand for videos of high production value. However, there are few tools that are tailored for producing high quality computer science videos. To address this gap, we have created an animation engine that can flexibly generate videos for the purpose of algorithms education.

Examples of algorithm visualization libraries include *VisuAlgo* [5], a web app which allows users to interact with a large collection of classical algorithms and data structures, and *The Sound of Sorting* [2], a web app that visualizes sorting algorithms with sounds. Video recordings of this app on *YouTube* [1] have been watched over 17 million times, which indicates the popularity of such visualizations.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576233

2 OVERVIEW / METHODS / RESULTS

We developed an object-oriented library called *AAnim* for creating animations of algorithms. *AAnim* is based on *Manim* [6], an animation engine for making basic animations and shapes originally developed for the "3Blue1Brown" YouTube channel. *AAnim* addresses two limitations of *Manim*: its high learning curve, and its lack of built-in functionality for animating common computer science constructs.

Users interact with *AAnim* through *input commands*. In particular, the user does not need to know how to use *Manim*. For a graph algorithm, the graph itself is the input. In the case of a data structure, these commands are a list of queries. Graphs are positioned using a layout engine in *Graphviz* [3], while other data structures, *e.g.* arrays and heaps, are given their usual layouts. Then, *AAnim* executes the list of queries or the graph algorithm, animating the data structure and pointing to the execution's current position in the pseudocode. Finally, *AAnim* generates a video file as the output.

To make the videos accessible to those with color vision deficiency (CVD or color-blindness), we set the color palette to be as easy-to-distinguish as possible. We tested its effectiveness using *Sim Daltonism* [4], an app which allows us visualize colors from the perspective of someone with CVD. In addition, we introduced at least one extra visual or auditory hint so that objects are not distinguished by colors alone, such as text labels, sound effects, etc.

3 FUTURE WORK

Currently, all input is manually entered, but for certain algorithms, we are working towards generating "interesting" random examples automatically. For example, a good input to Prim's algorithm would result in a minimum spanning tree with many branches, rather than just a single path.

One drawback of using *Manim* is that generating a new animation might not be feasible in a live setting, such as in a classroom. Its slow rendering speeds also impact production time, where a user might need to go through several iterations before being satisfied with the output. We would like to add real-time "preview" component to address these concerns.

REFERENCES

[1] Timo Bingmann. 15 Sorting Algorithms in 6 Minutes. https://youtu.be/kPRA0W1kECg, 2013.
[2] Timo Bingmann. The Sound of Sorting. https://panthema.net/2013/sound-of-sorting/, 2013.
[3] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen North, and Gordon Woodhull. Graphviz — open source graph drawing tools. In *Lecture Notes in Computer Science*, pages 483–484. Springer-Verlag, 2001.
[4] Michel Fortin. Sim Daltonism. https://michelf.ca/projects/sim-daltonism, 2020.
[5] Steven Halim. VisuAlgo. https://visualgo.net/en, 2011.
[6] The Manim Community Developers. Manim – Mathematical Animation Framework. https://www.manim.community, 2022.