IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 4, April 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004029 |

# Implementation of Visualizing Sorting Algorithms

**Shital Chatter[1], Sakshi Nalkar[2], Shweta Bhosale[2], Harshada Thite[2], Varad Chilkhalikar[2]**

Department of Computer, Pimpri Chinchwad Polytechnic, Akurdi, Pune, India[1]

PCP Students, Department of Computer, Pimpri Chinchwad Polytechnic, Akurdi, Pune, India[2]

**ABSTRACT:** This paper discusses a study performed on animating sorting al- gorithms as a learning aid for classroom instruction. A web-based animation tool was created to visualize four common sorting algo- rithms: Selection Sort, Bubble Sort, Insertion Sort, and Merge Sort. The animation tool would represent data as a bar-graph and after se-lecting a data-ordering and algorithm, the user can run an automatedanimation or step through it at their own pace. Afterwards, a study was conducted with a voluntary student population at Rhode IslandCollege who were in the process of learning algorithms in their Com-puter Science curriculum. The study consisted of a demonstration and survey that asked the students questions that may show improve-ment when understanding algorithms. The results and responses are recorded and analyzed in this paper with respect to previous studies

## I. INTRODUCTION

How do you work out a problem? The problem itself doesn't need to be anything overly complex, such as trying to replace a broken headlight in your car (although nowadays, manufacturers are trying the patience of the community with their increasingly abstract, space-age designs). The point is how to *attack* the problem. Do you perform research, such as looking through your car's manual for step-by-step instructions, or is your first instinct to findsomeone who knows how to do it (whether they are right next to you or inan online video)? My instinct is the latter, as I am a visual learner and amadept to picking up concepts by *seeing* it done, rather than *reading* about it.For example, when I was learning about sorting algorithms while pursuing my Computer Science degree, I found that *seeing* the data move to its correct position under the constraints of an algorithm was much easier to follow thantracing the code by hand.

## II. LITERATURE SURVEY

The four algorithms are: Selection Sort, Bubble Sort, Insertion Sort, and Merge Sort.Keeping in line with the example of sorting people by age, let's pretend that you have printed the age of eachperson on a separate index card. Oneway to go about organizing the cards is to first find the smallest age inthepile and bring it to the front. Then, find the next smallest and place it behind the already ordered first age.Eventually, you will end up with a pileof index cards that list the ages in ascending order. This method is exactlyhow Selection Sort works, where to sort a set of data, you select the smallest first, and then the next smallest and the next smallest. This algorithm is not very difficult to understand by word of mouth, but more abstract sorting algorithms, such as how Quick Sort requires moving data around a pivot point, may not be intuitive to read through.

## III. PROPOSAL

The organization of the code follows both object-oriented and functional programming concepts. Originally, the design was almost completely func- tional, where only three objects were used: one to control the canvas that displayed the animation, another to represent a piece of data, or "bar" object (blue rectangle with dynamically changing height and position), and a final one to represent the positions that each bar moved to, or "pos" objects. Some instance variables and Boolean values were used to keep track of the algo- rithm selected and when to animate, but this resulted in a heavily integrated mass of function calls that was difficult to upkeep.

## IV. RESULTS AND DISCUSSION

The best way to go about using the tool is to first select the ordering ofthe data and then select which algorithm to visualize. When any one of the algorithm buttons are selected, it will sort the data as it appears on theinterface. The ordering takes precedence, as selecting the ordering after the algorithm updates the interface momentarily, while the code has already run the initialization with the previous data set. After conducting the surveys,this sparked some

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004029 |

confusion as the algorithm buttons are listed above the ordering buttons in the interface. One student commented on having diffi- culty trying to start sorting, thinking that it may be the cause of pressing the buttons in the wrong order, which in turn did not run the animation.

## V. SIMULATION RESULT

One large refactor later, the code now resembles a Model-View-Controller architecture. Although, due to its functional nature, it has many more indi- vidualized functions that update the instance variables and Boolean values, thereby directly updating the View and Controller. A simplified diagram of the Model- View-Controller relationship is below in figure 8

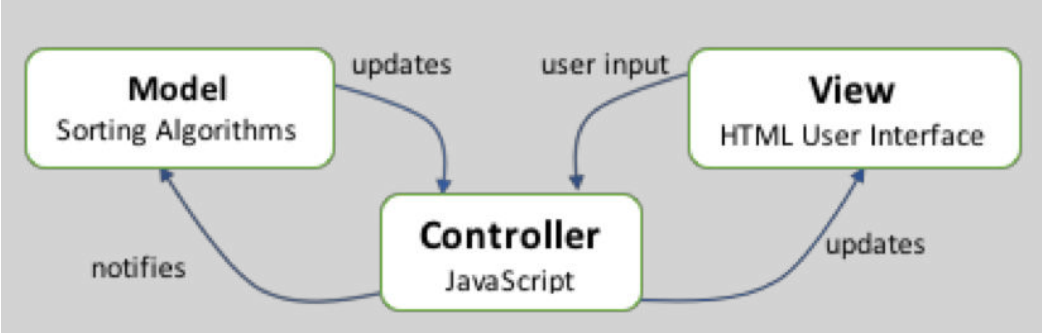


Figure8. Model-View-Controller diagram of code

- Selection Sort : It is a simple sorting algorithm.This sorting algorithm is an in place comparison based algorithm in which list is divided into two parts , the sorted part at the left end and the unsorted part at the right end. Initally , the sorted part is empty and the unsorted part is the entire list.

- Bubble Sort : Bubble Sort is a sorting algorithm that compares two adjacent elements and swaps them until they are not in the intended order.

- Insertion Sort : Insertion Sort is a simple sort algorithm that builds the final sorted array one item at a time. It is much less efficient on large list than more advanced algorithms such as quicksort, heapsort, or merge sort.

- Merge Sort : Merge sort is a sorting technique based on divide and conquer technique. Merge sort first divides the array into equal halves and then combines them in a sorted manner.

## VI. CONCLUSION

Through much time and effort, I have successfully created a working web- based animation tool for visualizing the following sorting algorithms: Selec- tion Sort, Bubble Sort, Insertion Sort, and Merge/Insertion Sort. Even with its memory overhead, it received overall positive feedback from the students who explored it. I am not surprised that there was not a significant differ- ence in learning the material, which reflects what I found in my previous research. There remains, however, a strong mindset to research and create animations like these to improve learning in the classroom, which I agree with completely.Learning how to code a web platform was challenging, and I thank thetutorials on W3Schools.com for getting me there. I had a previous internship where I updated the JavaScript on a webpage, but it was much more concise and did not involve objects and HTML for visualizations. The good newsis that JavaScript is still one of the most popular web languages, so I am not too worried about another big refactor soon for a language update.

## REFERENCES

1. Bingmann. “The Sound of Sorting - ‘Audibilization’ and Visualization of Sorting Algorithms.” *Panthemanet Weblog*. Impressum, 22 May 2013. Web. 29 Mar. 2017. <http://panthema.net/2013/sound-of-sorting/>.
2. *Bubble-sort with Hungarian (“Cs´ang´o”) Folk Dance*. Dir. K´atai Zolt´an and T´oth L´aszl´o. *YouTube*. Sapientia University, 29 Mar. 2011. Web. 29 Mar. 2017. <https://www.youtube.com/watch?v=lyZQPjUT5B4>.
3. A. Kerren and J. T. Stasko. (2002) Chapter 1 Algorithm An- imation. In: *Diehl S.(eds) Software Visualization*.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004029 |

Lecture Notes in Computer Science, vol 2269. Springer, Berlin, Heidelberg. <http://homepage.lnu.se/staff/akemsi/pubs/22690001.pdf>.

4. as engaging learning tools. *Proceedings of the Koli Calling '07 Proceedings of the Seventh Baltic Sea Conference on Computing Educa-tion Research - Volume 88*, Koli '07 (Koli National Park, Finland), pages203-206. <http://dl.acm.org/citation.cfm?id=2449352>.
5. J. Stasko. Using Student-built Algorithm Animations As Learning Aids. *Proceedings of the Twenty- eighth SIGCSE Technical Symposium on Com- puter Science Education*. SIGCSE '97 (San Jose, California), pages 25-29. <http://doi.acm.org/10.1145/268084.268091>.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details