Summer Internship Project Report

Visualisation of data structures in ELM

Submitted by

Shivankar Pilligundla

Shivankar.Pilligundla@iiitb.ac.in

Undergraduate student
IIIT Bangalore.

Under the guidance of

Prof. Venkatesh Choppella

Software Engineering Research Center
IIIT Hyderabad

Summer Internship 2022

Contents

0.1 Introduction

This report outlines my work as part of my summer internship at the Software Engineering Research Center, IIIT Hyderabad from 15th May-30th June 2022. I have been working with the Algodynamics team to build a data structure visualization package in ELM. This package abstracts the parameters required for drawing/rendering a data structure and provides APIs which can be directly used to render a data structure like Arrays, Stacks, etc on to the screen.

0.2 Objective

1. Understand Algodynamics Experiments.
2. Learn ELM language.
3. Learn Modelling of a problem in ELM.
4. Extend elm-dagre package and publish a package which supports drawing of other data structures like Array, Stacks, Lists, etc.

0.3 Work Done

0.3.1 Implemented Bubble Sort replica using ELM

I have learnt elm by starting with implementation of bubble sort experiment Replica. This helped me learn to model a problem in ELM and understand language constructs, architecture of ELM. My implementation has the following features:

1. Displays and Updates state variables namely index(i) and boundary(b) of bubble sort Experiment.
2. Highlights elements with green color if they are in the correct position when compared with sorted Array.
3. Reset button to reset the machine to initial state.
4. Undo button to undo the previous operation.
5. Status string which displays array status i.e sorted or not sorted.

6. Swap button to swap consecutive elements.

7. Index highlighter which highlights indices of two elements which are going to be swapped.

This can be tried out by visiting the following link:
bubble-sort link
Repository Link:
Repo Link

0.3.2 Elm Calculator

I have implemented a fully functional calculator in ELM. This helped me learn how to write logic/algorithm for a particular task in ELM and how to divide it into sub-problems and use function composition in ELM. This resource helped me in understanding function composition better

function composition

This Calculator has some drawbacks like:

1. It does not have BODMAS Functionality for calculations.

2. It is not configurable and layout has been designed by hard coding the coordinate values and sizes.

This calculator can be accessed from the following link:
elm calculator
Repository Link:
Repo Link

To overcome these drawbacks I have implemented configurable calculator which is described in the next section.

0.3.3 Configurable ELM calculator

This calculator is built to overcome the drawbacks previously implemented version had. It has the following features:

1. It has BODMAS Functionality for calculations.

2. It is highly configurable.

3. Its layout is calculated dynamically i.e height and width of the calculator is calculated based on number of buttons.

4. We can choose no of rows and columns to be present.

5. Each button is configurable with parameters like shape, size, fill color, stroke color, stroke width, etc.

6. The displayed text on buttons and screen is customizable with different fonts and sizes.

The calculator can be found at the following link:
Configurable Calculator
Repository Link:
Repo Link

0.3.4 ELM data structure package

We have published a package called elm-datastructure(link). This package is extension of another package called elm-dagre(link). Currently this package supports drawing/rendering view of the Array data structure.

The primary objectives of the package are as follows:

1. To design and implement a custom drawer function in pure Elm which takes in a user defined array as a parameter and outputs a clean representation of it in an SVG container.

2. To make the visualizing of an array seamless and minimalistic, without testing the users proficiency in graph-related attributes with the help of abstraction.

3. To allow the user to add their own custom attributes to modify the rendered output, without any inconvenience.

0.4 Future Work

As the name of the package(elm-datastructure) suggests, our plan was to implement a package which supports drawing of all possible data structures. But, we were only able to implement array-view in teh current version due to the time constraints. The further plans include:

1. Add more data structures to the package like Stacks, Queues, Lists.
2. Add bar graphs and histograms to Array-view so that algorithms like bubble-sort can be visualised easily.
3. Write tests to the current package using elm-test.
4. Make changes to the package to add responsiveness, so that it could also be rendered on mobiles.

0.5 Additional Links

0.5.1 Internship work demo presentation:

PPT link

0.5.2 WorkLogs:

worklogs link

0.6 Acknowledgment

I would like to thank our mentor Archit Goyal for guiding us throughout the internship. I would also like to thank my teammates Rohail Alam and Mrityunjay Jha.

0.7 References

[1] guide.elm-lang.org

[2] elmprogramming.com

[3] package.elm-lang.org/packages/goyalarchit/elm-dagre/2.0.0/Dagre

[4] korban.net/posts/elm/2018-10-02-basic-steps-publish-package-elm-19/

[5] medium.com/@Max_Goldstein/how-to-publish-an-elm-package-3053b771e545