|  |  |
| --- | --- |
| Date | Activity Log |
| 10 June 2021 | Today I came up with the first iteration of my Design Brief. I used [this](https://www.freecodecamp.org/news/sorting-algorithms-explained-with-examples-in-python-java-and-c/) article I found the sorting algorithms I will be using. I found the searching algorithms [here](https://www.geeksforgeeks.org/searching-algorithms/) and the pathfinding algorithms [here](https://www.geeksforgeeks.org/fundamentals-of-algorithms/?ref=shm). I then created three mermaid graphs to show all these algorithms. |
| 14 June 2021 | I started work on my project application and wrote the introduction to my report. I decided to use Facebook react for this project. I do not know how to use Facebook React yet, so I will have to learn. I have chosen React because it allows you to reuse components which will be very useful. |
| 21 June 2021 | Today I started an online course on react.js, which can be found [here](https://teamtreehouse.com/tracks/learn-react). This, alongside some further research, should hopefully be enough for me to carry out the project. I learned how React breaks down a website into reusable individual components that are then rendered into HTML elements. These components can change based on data in a javascript file. This will be very useful for my project. For example, in my sorting visualiser, the elements will be able to change their own height based on their value. |
| 24 June 2021 | Today I decided on what to name my website. I subsequently purchased the domain visual-algo.tech, which will be used for the project. I will host the website on GitHub pages. This is because it is both free and easy to update as I will store the project on Github. |
| 1 July 2021 | Today I continued learning about Facebook React. I learnt about react props which allow you to reuse React components by passing different data (Props). The problem with props is that they are not changeable. This issue is addressed with react states. States allow you to change the components data dynamically. To use states, you have to convert all your react components to classes. I completed the react basics of the course. It made me realise how useful react could be due to how it works. I', considering using typescript instead of javascript for my application. Typescript is a superset of javascript. However, it contains the added features of type checking. This is especially useful when it comes to debugging code, as errors can be found when the typescript code is transpiled into javascript |
| 4 July 2021 | Today I played around with a prototype for the sorting, It was my first time using react, and upon using it, I realised how useful it would be. I had a look at [this website](https://algobook-io.github.io/Searching-Sorting-Visualizer/#/sort), and I liked how the bars moved across the scene. I had a look at some of the code of the website to try and figure out how it is done. I discovered that they used CSS transformations with a translateX property and an animation delay, I incorporated this into my testing, and I was delighted with the result. It is definitely something I will incorporate into my application |
| 5 July 2021 | Today I wrote up my Project introduction. after some thinking, I played with the idea of changing what I am doing for this project. This was because I was worried the current project would allow me to achieve a good grade. Therefore I thought about instead using machine learning to generate my own holy scriptures, which would be created from learning from the bible. After thinking about it a bit more and after talking to my supervisor about it, I decided to stick to my original plan. This was for two reasons, one being it would be a lot of work to change now. the other being the subject area is probably too similar to my computer science NEA |
| 8 July 2021 | Today I started creating my artefact. After some further research into the subject and watching [this video](https://youtu.be/ydkQlJhodio), I decided to use Typescript. This is because it allows for greater error checking due to its usage of static types. However, since it is a superset of javascript, all javascript is valid typescript code. I then watched [this video](https://youtu.be/ahCwqrYpIuM) to learn the basics. |
| 11 July 2021 | After continuing with my artefact today, I encountered an issue. I realised the way I was planning to swap around bars would work. However, it would have the caveat of making it difficult to change the height of the bars (required for making it work on smaller devices). I originally planned on having all the bars start off on top of each other, then offset the bars relative to their position in the array. That way, all I would have to do to swap two bars is switch their position in the array. The problem with this is that in order to have the bars on top of each other at the beginning, you have to set their position to absolute, meaning that they are on top of everything and independent. This means that the height of the bars cannot be determined based on the height of their container. Whist this could be fixed, it would make things a lot more complicated and slow. The alternative method I came up with was to start each bar at a random position, then instead store an offset variable. Then, when you want to swap two bars, you add or subtract their offset values to swap their positions. This will mean that the heights of the bars will be able to be set as a percentage of its container, fixing the issue |
| 11 July 2021 | It turns out my alternative idea, whilst fixing the issue of dynamic height, creates a new issue of making it so that you are unable to have dynamic width. After another look at [this website](https://algobook-io.github.io/Searching-Sorting-Visualizer/#/sort), I spotted that it also has this issue when you go to smaller screen sizes. This is due to how flexbox squishes things together when their combined width is greater than the container width, making it impossible to know the exact width of bars and swap bars with an offset exactly. Therefore, I have decided the best way to do this is to only swap the heights of two bars rather than their positions. Whist this won't look as good as if the bars swapped position, it will mean that the application will be able to have dynamic width and heights. It will also simplify the code for sorting the bars. |
| 12 July 2021 | Today I continued on the artefact. I added in the bars and their ability to be generated based on the value inside the range input. I added the bars for the visualiser using my revised way of doing it and the ability to select how many bars there are via a range input slider. I also started to implement the algorithm selection and the framework for running the algorithms. |
| 13 July 2021 | Today I finished the algorithm selection. Whenever the user selects an algorithm, a method is called which runs the algorithm, and the steps involved are recorded. An issue I has was where I didn't sterilise the current state of the bars when they were appended to the steps taken array. This caused them to all be linked, and changing one would update all of them. I only realised this by adding the steps into the application state. Doing so would update the bars on the screen, which confused me as I had not changed their values. I spotted the issue on closer inspection and rectified it by converting the data into JSON and back again. |
| 14 July 2021 | Today I wrote about and implemented the first algorithm, Bubble sort. I chose to do this first as I am familiar with it from learning about it in Computer Science and Further Maths. I used a mixture of my own prior knowledge and [this](https://en.wikipedia.org/wiki/Bubble_sort) Wikipedia article to help me. I rectified an error with my method of calculating the steps every time an algorithm was selected or when the bars were modified as it was vastly inefficient. My new approach is to calculate it only once sorting has started. I am uncertain how I will present my sorting algorithm research as, at the moment, it is in the middle of my code as the code from it is required. However, I would like an algorithm section at a later date once I have researched the rest of the sorting algorithms. For now I will focus on getting the rest of the sorting section functional before I add in the rest of the algorithms |
| 15 July 2021 | Today I got the visualiser to visualise the bubble sort algorithm I had implemented. I almost went down a whole different direction with how the visualisation would work. The function setState runs asynchronously, which resulted in the usRunning variable inside the state not being set to true before the visualisation stated, causing it not to start. Unsure how to fix this, I had a look at how [this visuliser](https://bennettoh.github.io/sorting-visualizer/) dealt with this and saw that they used timeouts. However, instead of rewriting my visualising code to work in that way, I realised I could just put await before the setState, which fixed my issue. With the visualiser then working, I added the ability to adjust the speed at which it visualised |
| 17 July 2021 | Today I added buttons to step forward and backwards through. These buttons were not in my original plan for the sorting page; however, I decided to add them as they allow the user to step forward and backwards through the visualisation at their own pace. |
| 19 July 2021 | Today I added Colour changing to the visualisation. I used my first custom type for the state of the bars. I found the ability to see the four different options on the intellisence very useful |
| 20 July 2021 | Today I added the Measurement of swaps and comparisons to the visualiser. Due to the addition of the \bility to step forwards and backwards, measuring the time is no longer doable. Therefore I had to remove it. I also reordered all my states and interfaces into a new file so that they are easier to manage |
| 21 July 2021 | Today I added the description. I initially added it as a fullscreen popup window. However, I wouldn't say I liked how that looked, so I changed it just to be added to the bottom of the webpage instead. I might return to that way of doing it once I have decided on the website's visual look. I also added a progress bar to the visualiser to show how far through the algorithm it is. I'm considering changing the range slider for the number of bars and speed to be a dropdown menu instead. This is because you do not need to be able to select the number of bars exactly, and the speed slider is non-linear due to how the delay is calculated, leaving any option less than 75% of the speed all looking the same. I am still undecided on the visual design of the website however I drew some quick mockups to go through some ideas |
| 22 July 2021 | Today I split up my Control react component down into components for the individual buttons. I did this because it will allow me to have more excellent reusability and manageability. |
| 19 August 2021 | Today I created a navbar for the application. I originally planned on having this bar along the side of the screen; however, after experimenting with it, I decided it was not the way I wanted to go as it left too much blank space, and you could not have text without expanding it. Therefore I went with a top bar design instead. I also decided on the Nord colour scheme for my project. I encountered this colour pallet when setting up and theming my arch Linux installation on my laptop based on [this image](https://i.redd.it/y6ggh8x53yb61.png). This pallet will allow me to implement both dark and light interfaces in the future. By forcing myself to use these 16 colours, it will make the look of the application be more consistent and appealing |
| 20 August 2021 | Today I modified my navbar so that it works on mobile devices. Before I did this, once the screen width became too small, the navbar went over the sides of the screen and became unusable. because this is an educational website, it needs to work on mobile, so dropdown menu now appears instead of navbar links when the screen size reaches mobile levels |
| 21 August 2021 | Using the colours I had defined before, I started adding some theming to the interface. I used a CSS grid to arrange all the UI elements. I also created some new elements to show what the colours of the bars represent. This is in order to |
| 30 August 2021 | I added more styling along with a key detailing what the colour of each bar represents. This will improve the ease of use of the program. |
| 6 September 2021 | I finished off the styling of the application. The application now has a consistently clean and modern interface using colours from the nord colour pallet. This also extends to the mobile version. I fixed a bug with the mobile version where the control boxes were stacked on top of each other. The issue was that the CSS file containing the rules for how they should align was not being imported correctly, resulting in the desktop alignment being used. |
| 7 September 2021 | I contacted my EPQ supervisor with concerns about the project and if it was sufficiently complex enough to get a high mark. I proposed an alternative option where I use artificial intelligence trained on the bible to generate new scriptures. He liked the sound of my idea, however recommended I stuck to the current project. |
| 28 November 2021 | Due to my time being taken up by my university application, entrance exams, and my computer science coursework, I have been unable to complete much work on this project. Today I implemented the quicksort algorithm. It was a challenge to implement a recursive algorithm visually, however I figured out a way to keep the process visually understandable by only tracking the local stages, then combining those with the stages returned by any recursive calls. This then results in the final stages array containing all the stages in the sort in the order they were executed |
| 4 January 2022 | Today I implemented insertion sort. It was relatively simpler than quicksort as it does not involve recursion, however I did struggle with how to colour code it. Due to the way it works most bars won’t be in the correct position until the end, therefore I set all bars inside the sorted section to green rather to make it easier to distinguish the two sections |
| 6 January 2022 | Today I implemented merge sort. Merge sort was the most difficult to implement, which is shown by the length of the code. This is reflected in the length of the finished algorithm. A lot of trial and error was required in order to correctly swap items. I could have done it in a simpler method using multiple arrays, however that was not as good to visualise. I also fixed a bug where the dropdown menu to select the algorithms was not changing it’s size. |
| 7 January 2022 | Today I implemented selection sort. I also fixed a bug in the speed selector where the initial speed was not the speed displayed on the selector. I also doubled the speed of the bar swapping animation, as I found it to look to slow. Another thing I did was when the selected algorithm is changed, the stages are set to the beginning, allowing the ability to compare multiple algorithms on the same data. All the sorts for my application are now implemented |
| 10 January 2022 | Today I added descriptions to all the algorithms I have implemented. I based them of what I had written for the research report, however without any referencing. I also went back and cleaned up my algorithm code. |