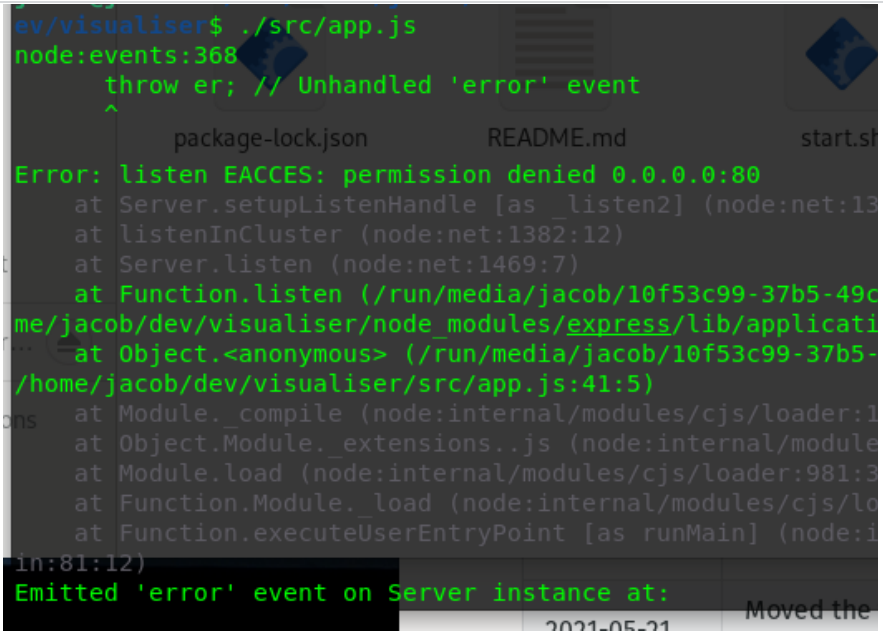
NEA Project Log

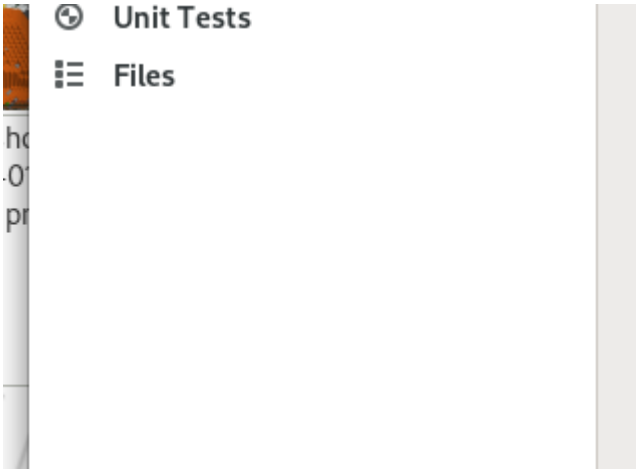
Jacob Halleron
The Sandon School

Date	Note
2021-03-22	Created analysis document.
2021-03-25	Wrote “market research” section of analysis document. Emailed my client about the details of the project.
2021-04-01	Initialised git repository so that my code isn’t just stored on my laptop. Started using git commit messages rather than a word processor for the project log.
2021-04-04	Created “readme” file for GitHub.
2021-04-19	Updated the project analysis document by writing the proposed solution section.
2021-04-22	Initialised NPM so I can run JS files with imported modules. Also installed Express, the web server module.
2021-04-22	Created basic web server along with index.html that links the other pages, 404 page and a test page which I’ll use to test the bar chart visualiser. Also created basic stylesheet.
2021-04-22	Tried to run git push but the previous commit was on a school computer, so my laptop’s local repository was behind the online repository. I had to use git pull, then commit and push.
2021-04-22	**Problem:** Web server won’t start up. The code looks pretty boilerplate, and I expect “server alive” to be printed to the console at startup, but instead I get this error message:

Date	Note
	 **Solution**: I realize I've set the server to listen on port 80 (for HTTP), which requires root privileges to access. I'll make sure to write `sudo ./app.js` rather than just `./app.js` in future.
2021-04-26	Moved web server address from localhost:8080 to localhost:80 so that it doesn't interfere with my other programs. I should stop coding for now, I don't have proper documented design for the project.
2021-05-06	Added model section to analysis ODT.
2021-05-21	Moved the to do list to the proposed solution section and wrote the objectives section based on emails with the client.
2021-05-24	Tweaked the to do list, specifically the site design.
2021-05-27	Added git ignore file so that git doesn't upload the relatively chunky node_modules directory to the online repository.
2021-06-17	Started work on the project design document. Designed some of the bar chart class.
2021-07-17	Implemented a crude function that draws a bar chart. I haven't yet encapsulated it into a visualiser class so it's pretty messy for now. Also added bogo sort, the simplest algorithm I could think of.

Date	Note
2021-07-29	Tweaked the CSS and added a few buttons so that I can (mostly) interact with the page without relying on the console.
2021-08-06	Added bubble sort and a visualiser class. Also added a list in the HTML that gets automatically updated with the variables of the currently running algorithm. The whole thing is still very buggy.
2021-08-08	Didn't add any new features, but I did clean up the code. Encapsulated most functions and variables into the SortChart class. I'm naming the classes "charts" rather than "visualisers", I think it's easier to read.
2021-08-23	Made sure the getters and setters of the SortChart class all made sense and worked as one would expect. Also added the `interval` property so that the speed can be changed.
2021-08-24	Added insertion sort. Also added sound effects so that the page beeps and boops for different heights of bar. It isn't appropriately distributed using the complex maths yet.
2021-08-24	**Problem:** Trying to make sound effects work. I expect there to be a smooth stream of legato tones of varying pitch and currently, they exists but sound choppy and disjointed. Every time the displayed pointer moves, a new oscillator is created that plays the sound and is deleted. **Solution:** I'll make one oscillator for each chart on initialisation, but set its volume to zero. Every time I need a sound, I'll make the volume high enough and give it the correct pitch.
2021-09-06	Started on the tree and graph drawing algorithms. Created a HTML and JS file for each, similar to the bar chart. It should be easier since I've already figured out the groundwork, the main difference will be the draw function. The bar chart is very slow for larger inputs, the time for one cycle is capped at about 5ms. I'll need to optimise the script so that I run the drawing function every frame (60Hz) rather than every time I update the array to eliminate pointless unseen drawing.
2021-09-06	**Problem:** Added a tree class but the method to recursively generate branches only creates some of the tree. Here is the current output: It returns a tree that is 3 nodes deep,

Date	Note
	but only contains 4 nodes rather than 13. **Solution:** The problem is that the `i` from the method's for loop is not reset when the function is called again. I forgot to use the `let` keyword before the `i = 0` to make it have function scope rather than global scope.
2021-09-07	Added more comments for better code readability and tidied up the design document with all the methods and variables in tables.
2021-09-09	Added small startup script for operator efficiency. It should save a few seconds whenever I want to run the server in future.
2021-09-12	Optimised bubble sort and added "boggle sort", a made up combination of bogo and bubble sort because it seemed fun. In the end I'll want the graph, tree and barchart classes in the same file for better load times and simpler layout, so I've added a basic "Chart" parent class and started moving the methods over.
2021-09-26	Moved the project log back from the git commit messages to this document using a word processor, it will look more complete when the project is done.
2021-09-28	Explained my algorithm code format better in the design document, I think it makes sense now.
2021-10-11	Added substantial comments to all scripts, documents and the style sheet.
2021-10-18	Planned the tree drawing algorithm in the design document. It turns out to be trickier than expected to generate coordinates for nodes in a tree, so I'll be using the "Reingold-Tilford algorithm" which I found in a blog post from 2014 drawing-trees/
2021-11-07	Restructured the directories so that all the code is in `src` and all the meta documents are in the `doc` folder. This is how most other projects I've seen are laid out.
2021-11-07	**Problem:** All code seems to have disappeared. I expect there to be a folder named `src`, maybe I accidentally deleted it? I hope not. **Solution**: I forgot to run `git add *` earlier when I restructured the folders, so git missed that

Date	Note
	one out last time I uploaded to the repository. 
2021-11-16	Properly implemented the logging so that it writes to a file as well as the console.
2021-11-16	**Problem:** The tree drawing algorithm requires that I access a tree node's siblings, but this is impossible with my current methods. I can only access a given node and its children. **Solution:** I'll need to completely change my tree class to allow for this. Currently a node's children are a property of that node, and I'll change it so that the `children` property is only an array of references to other nodes. Basically, trees will now be a subclass of graph.
2021-11-16	**Problem:** I can't manually edit the log file, my IDE just gives an error. **Solution:** Another permissions error. The log file was created by the server program, and the server program was run as a superuser, so I didn't have privileges. I sudo deleted it and manually made a new one.
2021-11-17	Changed the tree class so it's more graph-like, and moved all classes into the classes.js file.
2021-11-24	Added support for multiple charts at once.
2021-12-05	Changed the UI so that new charts can be added by the user and updated the CSS so everything is nice and centered.
2021-12-09	Chart now shows the current state of the running algorithm in the form of local variables.
2021-12-17	I've decided to trim down my targets for the project as I think I've given myself too big of a project for this, it won't be achievable in time. I'm not going to complete the tree or

Date	Note
	graph drawing algorithms that would represent binary search, Dijkstra's, etc. since I think these are relatively unnecessary and I can reach my objectives without this. I'm not scrapping the code though, it's just getting moved to a separate file.
2021-12-19	Implemented cocktail sort. This one's just bubble sort, but every other pass is in the opposite direction, so I just had an extra boolean variable for that. I could probably optimise it, but all the examples I've seen online aren't, so I won't.
2021-12-23	Restructured the server program. Instead of having pages for different types of chart, I've got a page to show one animation and a page to show multiple. This is more like how I designed.
2021-12-28	Tweaked the CSS. It didn't really alter anything programmatically, but now it looks a bit sleeker. I like it.
2022-01-02	Restructured the imports again. This time the way it works is the HTML file imports the main script, `rectangle.js` using a `<script>` tag, and that script uses the native JS import keyword to retrieve all the algorithms themselves, and their data. The object it actually imports is just a big map, which may be a crude solution but it works. Each algorithm's name maps to an instance of the algorithm class, which contains its functions and the type of algorithm it is (usually "sort")
2022-01-03	Redid the UI. Now there's the start, stop and reset buttons at the top of the page and a couple of input boxes for the speed and size of the chart(s).
2022-01-05	Added selection sort. This one's probably the slowest algorithm that isn't a joke, it goes through the entire array to find the smallest value and puts it at the start, then repeats.
2022-01-05	Cleaned up the code so there's no more redundant bits of code that I removed, or commented out regions of code. It's substantially shorter now.

Date	Note
2022-01-06	Redid some of the HTML. Now, whenever the program wants to add to the document, it inserts HTML directly instead of copying another part of the page. This is much cleaner, but I've got a lot of HTML code in my scripts now.
2022-01-06	Added even more comments. I hope there's enough comments.