Outline your project and which extensions you are building on to the template. (400 words) 256

I decided to extend the Data Visualization Project as it offered the most opportunities for creativity. Most drawing apps tend to look quite similar, and even music visualization didn't offer much room for originality. I also wanted to take this as an opportunity to learn something new and challenging.

However, with more freedom comes more potential issues. Since I decided early on to conduct complex 3D simulations, I ran into limitations imposed by P5.JS, the browser, and even the JavaScript language itself on several occasions.

The extensions I'm planning to do are the following:

1. Improve the Template given Visualization by using more colors, labels and chart-type modifications
    - Color Scale for the Bubbles based on Proportion of Females/Males
    - Hover Effect for Bubbles showing the type of Job
    - Add Labels and Colors to Climate Change
2. Creating a Simulation of a Simple Fully Connected Neural Network
    - Creating a Simulation of a Neural Network processing handwritten Numbers with 1 input layer, 1 hidden layer and 1 output layer
3. Creating a Simulation of a Convolutional Neural Network
    - Creating a Simulation of a Convolutional Neural Network processing handwritten Numbers with 1 Input Layer, 1 Convolutional Layer, 1 MaxPool layer, 1 hidden layer and 1 output layer
4. A 3D Map of a big City with Statistics shown as Bar Charts
    - Crime Statistics based on different Districts for New Yor City
    - Adding Hover Effect for the Bars
5. A Choropleth Map of the world showing the medium wages
    - A world map that on hover of specific countries elevates them and shows the medium wage, outside hover medium wage differences are color-coded

This will require the following coding techniques and study topics

- JavaScript
    - Arrays, different kinds of loops, advanced Array Manipulations (copy, sort, map, reduce, includes…)
    - Classes and Constructor Functions
    - Recursive Functions
    - Generators
    - Global, Local Variables
    - Lambda Functions
- Math
    - Several Matrix Operations
    - Exponential and Logarithmic Operations
    - Different forms of randomization
    - Trigonometric Functions
- WebGL Development
- DOM Interactions
- Code optimization for performance
- Browser Specifics
- Debugging, functional and user interaction testing

For the 3D Map the most challenging aspect was getting the projection correctly, for the neural network simulations the most challenging aspects were getting NumPy replacements in JavaScript and fighting with performance issues because it is evident that P5.JS is not made for complex simulations.

Question 2

Discuss the progress you have made on your project so far. (400 words)

I did research on several different sites and completed entire courses, books and lectures (from Stanford, Cornell University etc.) to develop the necessary knowledge base for these extensions.

It was challenging to estimate the time required for each task, and I either underestimated (in the case of the 3D Map and NumPy Replacement) or overestimated the time for several tasks. In the end, this balanced out, and I was able to stay on track with the overall project. However, it once again highlighted the unpredictable nature of software projects. Therefore, it was a wise decision to plan

the entire project in an agile manner, incorporating ample buffer time instead of adding another complex extension.

For the general interface design, I used Figma to create wireframes and developed a basic idea using Miro.

I utilized Post-its on a digital board and a simplified architecture plan to get an idea about what I had to build, which I then expanded over time.

For the neural network simulations, I first created pseudo-code variants, which I then translated to Python and later translated these prototypes into JavaScript.

Since I opted to tackle the more difficult aspects first, I would say about 70% of the project is finished. The 2D chart improvements and the choropleth are relatively less complex.

The most challenging part that remains is making performance improvements to accommodate a wider range of browsers and devices. I've already had to make numerous adjustments so the project could run smoothly on my main device, a MacBook Air M1, as well as an HP Spectrum x360 (2017) which I used for testing. However, it lags considerably in Firefox which seems to be a well-known problem for WebGL Applications.

There is also a high possibility that the sandboxes used by graders or devices with less computing power might encounter the same issues. To address this, I increased the buffer for testing and performance improvements from originally 1 task into 2 tasks on different timelines of the project.

Complex simulations like this in P5.JS are very sensitive. Slight changes in the code, version changes, or the use of different graphics objects can easily break the code and make it very hard to debug. Just to give an example, one texture bug that occurred due to a switch from canvas to createGraphics took me three days alone to resolve. Therefore, the most crucial aspect is to account for all circumstances and plan ahead.

Question 3

Discuss how you will organise you time for the rest of the project. (200 words)

Your answer should be based around a Gantt chart or other time organisation process. You can easily make this in a spreadsheet program and export an image or take a screenshot for inclusion here. Any words on the diagram do not count towards the word count.

If you use a Gantt chart should include the total time of the projects in weeks. You will also need to divide up the parts of your program and submission into activities for allocating time to on the chart.

- Your timescales should include progress to this point as well as afterwards.
- You should discuss how you have decided on the time you have allocated to each activity and why you have allocated these amounts of time.
- You should make reference to your chart to clarify the points you make here.

For time management, I used a Jira board where I created tasks and subtasks for each part that needed completion, then organized them into epics, which I presented in a Gantt chart.

Each task includes an extensive description of what to do, potential caveats that could occur, and an approximation of how much time it would cost.

I used my own estimates of difficulty, time, and unpredictability to calculate story points for each task, and grouped tasks into weekly sprints.

During the phase leading up to midterm, I averaged around 25 story points per sprint. But for the finals, I reduced this to 15 story points and increased the buffer at the end to ensure there was enough time to test and adapt to unexpected circumstances.

My learning from the midterm was that I need to start testing earlier and allocate more time for it. I extensively tested on my main browser(Chrome) and device(Mac Book M1 Air) but postponed testing on different devices and browsers until the end, where it was too late to make dramatic code changes. Therefore, I moved this task to earlier sprints and decided to tackle the harder topics again first, which in this case is the performance improvements.

Question 4

List any external sources that you have actively utilised in your project. This should include:

- any code you have used from external sources directly
- any code that you have taken inspiration from but adapted and refined for the project (such as pseudocode algorithms or code pens)
- any online help forums you have taken code from (i.e. StackOverflow or library documentation)
- any third-party libraries you are using.

You do not need to include everything you have read or that has helped you. Only where you have used or adapted code that appears in your project

There was no code I could directly use in my project, as it's rather rare to tackle this without any math libraries, especially in JavaScript. However, there were many resources in Python from which I was able to learn and adapt for my code.

The book "Make your own Neural Network" by Tariq Rashid is the resource I drawn the most from as my implementation in nn.js is an translated JavaScript Version without using NumPy.

The Rest of the resources were more used as a general implementation idea.
https://github.com/amanchadha/coursera-deep-learning-specialization Andrew NG Course
https://nnfs.io/
Box Mueller Transform
https://gist.github.com/yishn/f6483ac4c6db268dfa43914b44fab8b1

For the 3D Map as well, there is no code I could use unadopted but there were many Stack Overflow Resources that helped me find the right code

For the 3D Map I created a new Stack Exchange Account and asked this question

https://gis.stackexchange.com/questions/461880/selecting-layer-for-processing

And used these resources for knowledge

https://wiki.openstreetmap.org/wiki/Mercator#:~:text=Most%20of%20OSM%2C%20including%20the,a%20Web%20Mercator%20on%20Wikipedia.
https://stackoverflow.com/questions/14329691/convert-latitude-longitude-point-to-a-pixels-x-y-on-mercator-projection
https://medium.com/@suverov.dmitriy/how-to-convert-latitude-and-longitude-coordinates-into-pixel-offsets-8461093cb9f5
https://stackoverflow.com/questions/54588308/longitude-latitude-to-osm-pixel-in-256x256-tile
https://stackoverflow.com/questions/47836326/latitude-and-longitude-to-map-tile-set

For Third Party Libraries I am using

```
https://cdn.jsdelivr.net/npm/@davepagurek/p5.buildgeometry@0.0.5/build/p5.buildGeometry.js
```

https://mathjs.org/docs/reference/functions/min.html only for one reshape function I could not fix in time