

Visualisation of Epidemics Simulation

Matthew Lakin

Supervisor: Hossein Nevisi

October 2024 - May 2025

Contents

1	Abstract	2
2	Introduction	3
3	Problem Domain	4
3.1	Minimum Viable Product	4
3.2	Stretch Goals	5
4	Methodology	6
4.1	Gantt Chart	8
5	Research	9
5.1	Existing Solutions	9
5.1.1	COVID-19 Dashboard	9
5.1.2	Plague Inc.	10
5.2	Visualisation	11
5.2.1	Covid-19 Data Sources	11
5.2.2	D3.js	11
5.3	Compartmental Models	12
5.3.1	SIR Model	12
5.3.2	Solution to the SIR Model	12
5.3.3	Improvements to the SIR Model	13
6	Technical Solution	14
7	Results and Analysis	15
8	Conclusion	16
9	Bibliography	17
10	Appendix	18

Chapter 1

Abstract

[WILL WRITE AT END OF PROJECT]

Chapter 2

Introduction

"We've actually invested very little in a system to stop an epidemic. We're not ready for the next epidemic."

- Bill Gates, 2015 [5]

In recent years, we have all become too familiar with viruses and epidemics. From Ebola landing in Europe and America in the 2010s after ravaging through Africa, to Covid-19 in the 2020s. New epidemics seem to be springing up at a rate never seen before [7] and I think there are multiple causes to this. For one, access to medicine has been a huge help in increasing survival rates for all sorts of ailments. However, this has caused complacency in the world of medicine and Antibiotic Resistance, where medicine becomes ineffective to bacterial infections, has become one of the most prevalent issues the human race will face in the coming decades. Being able to use existing data to predict how epidemics will spread and what

Going back to the 14th Century, the Black Death, or the bubonic plague was one of the worst outbreaks in history. It killed between 75 and 200 million people in a few years time which was 30-50% of the total population of Europe at the time [3].

Smallpox still remains as the only human-affecting virus which has been functionally extinct as a result of human intervention [2]. This was a result of a combination of mass vaccination, containment and tracking. In this paper, I wanted to focus on the tracking aspect of containing diseases and how it can be a useful tool to not only teach us about what happened previously, but to predict and forecast what could happen in the future.

This whole section will be rewritten at the end of the project. I will be able to give a more accurate introduction to the project once I have completed it.

Chapter 3

Problem Domain

3.1 Minimum Viable Product

I have identified the minimal requirements for this project to be successful.

1. The dashboard must show a map of the Earth.
2. The dashboard must visualise the COVID-19 data and show the number of global cases and deaths per week.
3. The map will render polygon and country data from a GeoJSON file.
4. The individual countries must be clickable.
 - (a) Clicking a country will send the user to a new page.
 - (b) The new page will show the number of cases and deaths per week.
 - (c) The new page will show a graph of the number of cases and deaths per week.
5. The dashboard must show a timeline of the epidemic.
 - (a) The timeline will be a slider.
 - (b) The user must be able to scrub through the timeline.
 - (c) The timeline will show the current week.
6. The user must be able to select data from a file to be displayed.
 - (a) The code will accept a clean JSON file.
 - (b) The JSON file will have the same format as the COVID-19 data.

3.2 Stretch Goals

For this project, the following aspirational requirements have been identified:

1. Zooming in on the map.
 - This is my most realistic stretch goal. It would require using a library to render the map and allow the user to zoom in and out.
2. Hotspots on the map.
 - This would require analysis of the data to find out where the rate of infection is highest.
3. Analysis of data with other data sources to search for correlations.
 - I am currently unsure what data sources I would use for this, but it would be interesting to see if there are any correlations between the spread of the virus and other factors.
4. Predictive modelling and forecasting.
 - This is my most ambitious stretch goal. I would like to be able to predict the spread of the virus using the data I have. This would require a lot of work and research into the field of predictive modelling. This would likely be a separate project in itself.

Chapter 4

Methodology

For this project I will incorporate a Waterfall model into my project since it will ensure my code and report as a whole is developed in a structured and ordered fashion.

The steps of Waterfall I will be following are:

1. Requirements

- This section requires producing and refining a set of requirements. I plan on having 2 groups of requirements: Minimum Viable Product requirements and Stretch requirements.
- I plan on refining all the requirements in this stage so I can stay on this path plan.

2. Analysis

- In this section, I will be doing research into the problems which are discovered by my requirement refinement. I will investigate papers and other relevant published sources.
- I also want to investigate node packages to represent my data graphically. (This would continue into the design section)
- This section will likely take up a large section of my report, and I may return to this section midway through the project.

3. Design

- I would be designing the layout and colour scheme of my application as it would appear in a browser window. It is worth noting that different size screens would render it differently, so I need to take that into account when planning.
- I am not using a database for this project, but I will be using JSON and creating a standard design for objects in JSON will be beneficial.
- I need to decide what values I use for graphs and what kind of graphs to use.

4. Coding

- I will be making a web application with node.js. This will require me to create a valid work environment and get the node runtime working on my local machine. I will likely use Docker for this since it is a very easy to work with tool perfect for this application.
- I will mostly be using TypeScript for this project because of type safety and native compatibility with web development. There will also be HTML and CSS for loading the TS canvas and aligning elements.
- Node.js also comes with node package manager which will also be useful for loading the custom packages I plan on using in this project.

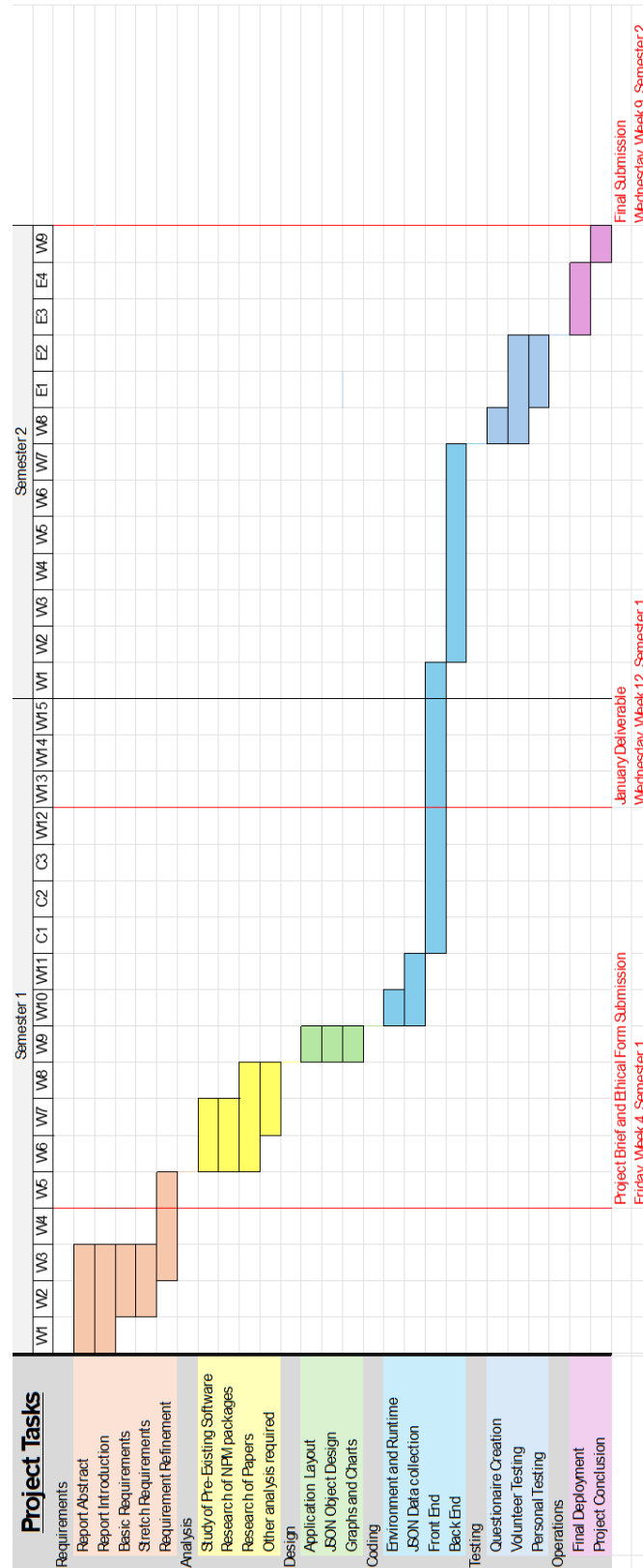
5. Testing

- For testing, I will be using a variety of normal, boundary and erroneous tests to check my software for crashes, unexpected results, and vulnerabilities. Using a type-safe language like TypeScript intends to minimise the chances of these happening.
- I also plan on having 3rd-party applicants to test my application. They would have a list of things to do in the application so I can analyse the ease of use.
- I would also ask them to fill in a questionnaire.

6. Operations

- I want to deploy my application to a server hosted by the university.
- Node.js applications are commonly deployed onto servers and having a production repository would be a good standard for myself. Pushing to a server from a local git repository is a highly transferable skill which is widely used. Test

4.1 Gantt Chart



Chapter 5

Research

5.1 Existing Solutions

5.1.1 COVID-19 Dashboard

A common tool for data visualisation is a software called PowerBI, a Microsoft product which allows for the creation of dashboards. The dashboard in the figure shows a real-time representation of the COVID-19 pandemic.



Figure 5.1: COVID-19 Dashboard Example

This dashboard has a lot of information on it even at first glance. The map uses bubbles to denote hotspots of activity for the virus. The bubbles are sized by the number of cases in that country. The bubbles can also be clicked for more detailed information on that country. The user is also able to alter the time frame of the data, and the data is updated in real time. There is a lot here which I would like to incorporate into my own project. The map is a good way to visualise the data and the bubbles are a good way to show the number of cases in each country. I also like the big bold text which summarises the data which is easy to read.

5.1.2 Plague Inc.

Plague Inc. is a game published in 2012 by Ndemic Creations. It gives the user the ability to create and spread a virus across the world. The game was initially released on mobile devices and has since been released on PC and console.

Mobile games require accessible and easy to use interfaces, which is why Plague Inc. is a good example of a visual epidemic simulation. The game is simple to use and has a lot of information on the screen at once.



Figure 5.2: Plague Inc. Dashboard Example

Despite many systems being dramatised for the sake of gameplay, the game offers many features which could be useful for a real-world epidemic simulation.

The game uses a world map which gradually changes colour as the virus spreads, this is from many red dots appearing on the map. The user also has access to individual breakdowns of each country and can see the number of cases and deaths in each country.

The game also has a transmission system which changes from country to country.

There is certain information provided in these examples that I will not be able to provide in my project. For example, the COVID-19 dashboard has real-time data, which I will not be able to provide. The Plague Inc. game has a transmission system which changes from country to country, which I will not be able to provide.

5.2 Visualisation

5.2.1 Covid-19 Data Sources

The data for this project will be sourced from the European Centre for Disease Prevention and Control (ECDC) [4]. The data is historic and only has data from the 1st week of 2020 to the 23rd week of 2022. The data is in a JSON format and includes the number of cases and deaths per week for each country. The format of the data is as follows:

Listing 5.1: Cases JSON Example

```
{
  "country": "Afghanistan",
  "country_code": "AFG",
  "continent": "Asia",
  "population": 38928341,
  "indicator": "cases",
  "weekly_count": 1368,
  "year_week": "2020-47",
  "rate_14_day": 6.5043,
  "cumulative_count": 44771,
  "source": "Epidemic
intelligence national data"
}
```

Listing 5.2: Deaths JSON Example

```
{
  "country": "Afghanistan",
  "country_code": "AFG",
  "continent": "Asia",
  "population": 38928341,
  "indicator": "deaths",
  "weekly_count": 69,
  "year_week": "2020-47",
  "rate_14_day": 3.3395,
  "cumulative_count": 1695,
  "source": "Epidemic
intelligence national data"
}
```

This is the format that the ECDC provide, but I will be amending it slightly to make it easier to work with. I will be combining the cases and deaths into one object and adding a date field. This will make it easier to work with the data in the front end.

It also needs to correspond with the GeoJSON data I will be using to render the map.

It is worth noting that I don't want this code to be limited to just COVID-19 data. I want to be able to use this code for any epidemic data. This is why I am making the data more generic and easier to work with.

5.2.2 D3.js

D3.js is a JavaScript library for producing dynamic, interactive data visualisations in web browsers. It makes use of the widely supported SVG format to render graphics. D3.js is a powerful tool for creating visualisations and is widely used in industry.

D3.js is a good choice for this project as it natively supports GeoJSON data which is the format I will be using for the map. [6]

This is exclusively used on a web browser, so I will be using node.js to serve the data to the front end.

The GeoJSON data has much more information than the COVID-19 data. It includes geometry of the country, information regarding the countries economy, sovereign status, GDP, all in multiple languages. I will be cleaning the data to only include relevant information for this project: country name in English, country code (to link the data with the Covid-19 JSON file) and geometry information.

5.3 Compartmental Models

5.3.1 SIR Model

Compartmental models are a type of mathematical model used to represent the different populations in a system. The most common compartmental system in epidemiology is the SIR model [1]. The SIR model is a simple model which divides the population into three compartments: Susceptible, Infected and Recovered. The model is represented by the following differential equations:

$$\frac{dS}{dt} = -\alpha SI \quad (5.1)$$

$$\frac{dI}{dt} = \alpha SI - \beta I \quad (5.2)$$

$$\frac{dR}{dt} = \beta I \quad (5.3)$$

Where:

- S is the number of susceptible individuals.
- I is the number of infected individuals.
- R is the number of recovered individuals.
- α is the rate of infection.
- β is the rate of recovery.

To explain this model, it's important to understand proportional reasoning. In the first equation, 5.1, the rate of change of the susceptible population is proportional to the complement of the product of the susceptible and infected populations.

A way to think of why that is true is because the more infected people there are, the more likely it is for a susceptible person to become infected.

This model is a good starting point for understanding the spread of a virus, but it has some limitations. In this model, once you have recovered, you are no longer able to be infected. Another issue is that the model assumes the disease is non-fatal.

5.3.2 Solution to the SIR Model

The SIR model can be solved using numerical methods. The most common method is the Euler method. The Euler method is a simple method for solving ordinary differential equations. It is not the most accurate method, but it is easy to implement. The Euler method is represented by the following equations:

$$S_{n+1} = S_n - \alpha S_n I_n \Delta t \quad (5.4)$$

$$I_{n+1} = I_n + \alpha S_n I_n \Delta t - \beta I_n \Delta t \quad (5.5)$$

$$R_{n+1} = R_n + \beta I_n \Delta t \quad (5.6)$$

Where:

- S_n is the number of susceptible individuals at time n .
- I_n is the number of infected individuals at time n .
- R_n is the number of recovered individuals at time n .
- Δt is the time step.

If we look at the phase space of the SIR model, we can see that the model has a fixed point at the origin. This means that the model will always return to the origin. This is not the case in real life, as the disease will eventually die out. This is because the model does not take into account the finite population size.

5.3.3 Improvements to the SIR Model

The SIR model can be improved by adding more compartments to the model. However, this makes the model more complex and harder to solve.

SEIR Model

This model adds an exposed compartment to the SIR model. For individuals who have been infected but are not yet infectious. The model is represented by the following differential equations:

$$\frac{dS}{dt} = \mu N - \mu S - \frac{\beta IS}{N} \quad (5.7)$$

$$\frac{dE}{dt} = \frac{\beta IS}{N} - (\mu + a)E \quad (5.8)$$

$$\frac{dI}{dt} = aE - (\mu + \gamma)I \quad (5.9)$$

$$\frac{dR}{dt} = \gamma I - \mu R \quad (5.10)$$

The SEIR model is better for simulations than the SIR model since it takes into account the incubation period of the disease. This is important as it is possible for a person to be infected but not yet infectious.

It is beyond the scope of this project to go into detail about the SEIR model, but it is worth noting that the model is more complex and requires more data to solve. The model is also more accurate than the SIR model.

Chapter 6

Technical Solution

Chapter 7

Results and Analysis

Chapter 8

Conclusion

Chapter 9

Bibliography

- [1] Ross Beckley, Cametria Weatherspoon, Michael Alexander, Marissa Chandler, Anthony Johnson, and Ghan S Bhatt. Modeling epidemics with differential equations. *Tennessee State University Internal Report*, 2013.
- [2] Centers for Disease Control and Prevention. Smallpox, 2023.
- [3] Sharon N DeWitte and James W Wood. Selectivity of black death mortality with respect to preexisting health. *Proceedings of the National Academy of Sciences*, 105(5):1436–1441, 2008.
- [4] European Centre for Disease Prevention and Control. European centre for disease prevention and control. <https://www.ecdc.europa.eu/>. Accessed: January 2025.
- [5] Bill Gates. The next outbreak? we’re not ready, March 2015.
- [6] Ash Kyd. Geojson vector maps. <https://geojson-maps.kyd.au/>. Accessed: January 2025.
- [7] Serge Morand, Muriel Figuié, Frédéric Keck, Claude Gilbert, Nathalie Brender, François Roger, and Patrick Zylberman. *Emergence of infectious diseases*. Quae Editions, April 2018.

Chapter 10

Appendix