Interactive 3D Map via OpenStreetMap

Melchizedek Gray

B.Sc. Computer Science

Lancaster University, England, United Kingdom

Abstract

Write abstract here

• Aims

• Design

• Implementation

• Findings

[Chapter 1. Introduction 4](#_Toc129544947)

[1.1 Aims of This Project 5](#_Toc129544948)

[Chapter 2. Background 6](#_Toc129544949)

[2.1 Overview of Existing Navigation Applications 6](#_Toc129544950)

[2.2 OpenStreetMap 6](#_Toc129544951)

[2.3 A-Frame 7](#_Toc129544952)

[Chapter 3. Technical Implementation 8](#_Toc129544953)

[3.1 Design of the System 8](#_Toc129544954)

[3.2 Overview of Languages and Software Used 8](#_Toc129544955)

[3.2.1 General Development Setup 8](#_Toc129544956)

[3.3 Discussion of OpenStreetMap Data Integration 9](#_Toc129544957)

[3.3.1 Getting the Data and Displaying It 9](#_Toc129544958)

[3.4 Explanation of 3D Rendering and Interactive Components 12](#_Toc129544959)

[Chapter 4. User Study 13](#_Toc129544960)

[4.1 Description of Research Questions 13](#_Toc129544961)

[4.2 Explanation of Methodology for Data Collection 13](#_Toc129544962)

[4.3 Analysis of User Feedback 13](#_Toc129544963)

[4.4 Evaluation of 3D Map 13](#_Toc129544964)

[4.5 Comparison of Map Features to Existing Solutions 13](#_Toc129544965)

[Chapter 5. Conclusion 14](#_Toc129544966)

[5.1 Recap of Key Findings and Results 14](#_Toc129544967)

[5.2 Discussion of How the Project Could Be Extended 14](#_Toc129544968)

[5.3 Final Thoughts on This Project 14](#_Toc129544969)

# Chapter 1. Introduction

For over two thousand years, humans have been practising cartography to create detailed maps [1] and without them, settlements would not have been able to communicate effectively, trade routes would not have been established, and navigation between countries over the sea would have been extremely difficult. Physical maps are helpful for navigating from Point A to Point B given that A and B are on the map with enough detail of the routes between. However, one major problem with a physical map is that as A and B get further away from each other the level of detail on the map diminishes greatly, e.g. villages between would be reduced to text. If you wanted to travel to multiple villages, towns, and cities and then explore those places, you would require multiple maps. This is where mobile phones step in. With their digital screens, they are able to show a map and have that map update as the user moves or interacts with it.

Considered to be the first mobile phone with Global Positioning System (GPS) built-in, the ‘Benefon ESC!’ was released in 2001 [2] and it paved the way for a new era of cartography. As GPS-capable phones became more widespread, some companies took notice of this new emerging market and capitalised on it and now, as of 2023, the total revenue in the navigation segment is over $1 billion [3] and is projected to reach an impressive $1.6 billion in 2027 [3].

Many companies in the mapping industry have strict licenses on who can access and use their data, such as Google Maps, which was made public in 2005 [4]. However, alternatives do exist such as OpenStreetMap (OSM). OSM is a freely accessible, community-driven mapping database that can be edited by anyone under the Open Data Commons Open Database License. By utilising its three main elements, nodes, ways, and relations, OSM allows for the inclusion of practically any geographical structure on the map, including, buildings, rivers, trees, bus routes, etc.

As the real-world changes, so too can a virtual map that uses OSM’s database. Due to its open data nature, anybody can update the map as they please, facilitating near real-time updates. This is extremely important for universities as they tend to constantly expand with new lecture, lab, and residency buildings. As an example, Lancaster University last erected a building in 2021. [5]. The ability to add that building to OSM’s database as soon as the doors open allows any mapping software using OSM to start displaying that building on day one.

In the recent years that we transitioned from paper maps to digital maps, the obvious evolution was to keep the format of a traditional bird’s eye view, especially due to the lack of performance available on mobile devices. This top-down view is excellent for understanding your whereabouts in a large area making it great for driving at speed as you can see upcoming turns easily. Though, with the walking speed of humans being considerably slower than a vehicle, we do not need to see nearly as far allowing for a more zoomed-in map. However, humans are not used to seeing the world from above and popular map applications such as Google Maps and Apple Maps do little with the extra zoom. Do I show a couple of example images here? Also, I feel like I’m being harsh and overall this paragraph I’m not that fond of. The lack of visual landmarks for the next turn and reliance on audio prompts creates a disconnected user experience, lacking a human touch.

A three-dimensional environment from the perspective of a human where the user can look around using their phone and easily identify buildings and paths and be able to visibly associate the next turn with a specific geographical feature would solve this issue. One benefit to this approach would be that the navigation feels more organic allowing the user to more intuitively get around. Another benefit would be that integrating the user’s experience in the world around them with the application allows for a more immersive experience.

Using OSM’s database to generate a 3D map of the Lancaster University campus

## Aims of This Project

This project has three overarching aims:

* Explore the use of open-data mapping databases to generate a 3D environment of the user’s current location.
* Investigate the effectiveness of a navigation system in that 3D space.
* Evaluate the implementation through a user study to understand the needs and wants of the users of such a system.

These aims will attempt to be met by using the A-Frame framework and OpenStreetMap’s open-data database to develop the map in a web browser and then conducting a user study on the implementation.

# Chapter 2. Background

## 2.1 Overview of Existing Navigation Applications

In terms of its user base, Google Maps dominates the mobile phone navigation market. A study completed in 2022 shows Google Maps as the fourth most popular application in the UK with an audience reach of 69% [6]. To give this some perspective, this is ahead of Facebook Messenger, Instagram, and Amazon, and with no other navigation applications below it for the next sixteen applications. Google Maps is quite the staple in navigation apps. It has grown from just navigating to being an all-in-one travel guide. It can suggest restaurants, suggest nearby hotels and how much they cost, and take you to sightseeing spots. It even shows customer reviews of restaurants and take-outs. This makes Google Maps quite difficult to compete against.

However, there are some navigation applications that have unique features, which have gotten recognition, such as Waze. In a study done in 2022 by AppMagic, Waze was the second most downloaded navigation app in the United States with 9.12 million downloads [7]. Waze is based around the community bettering the app's functionality for others by reporting certain things. For example, a person driving using Waze could report through the app that there has been an accident or that there is a speed camera. This will alert other drivers and even change their route if necessary.

## 2.2 OpenStreetMap

Founded in 2004 by Steve Coast, OSM was founded to give people, who face various barriers, access to free geographical data. Their database covers the entire world, including various geographical features such as buildings, shops, roads, trees, bins, and more. As mentioned earlier, OSM uses nodes, ways, and relations to label such data. A node is a single point on the map, a way is made up of a string of nodes, and a relation is a group of members that is comprised of an ordered list of one or more nodes, ways, and/or relations. A relation is used to describe a geographical structure between different objects, such as a lake with an island in the middle.

These elements can then be labelled with tags; tags consist of a key and value pair with there being over 90,000 unique keys [8]. For example, a node could have the tag “amenity=cafe” with the “name=…” tag denoting its name. A way could have the tags “highway=pedestrian” and “lit=yes”, this indicates that the way is a pedestrian path and it is lit with streetlamps. A closed way (a way that forms a complete loop) could have the tags “building=university” and “amenity=library”, to indicate that it is a university library building. Tags can be invented and added by anyone if there is no existing tag scheme for what is wanting to be added. Using these three basic elements with thousands of well-established tags and the ability to add tags, practically anything can be included in OSM’s database.

The collaborative side of OSM can be a major benefit but also a major downfall. This way of doing it relies on the trust of the community to keep the data accurate. If 1% of the users mistakenly or purposefully make errors, the other 99% are relied on to correct those errors.

## 2.3 A-Frame

A-Frame is a web framework for developing interactable 3D environments. It is based on top of HTML and three.js and uses a powerful Entity-Component System (ECS) that implements the composition over inheritance and hierarchy principle. An entity is comprised of one or more components that describe it. For example, a box entity may have a position, geometry, and material components. A couple of the benefits of an ECS, as listed on A-Frame’s website [9], are a greater flexibility when defining objects by mixing and matching reusable parts – this relates to using A-Frame’s mixins where the same material component could be used across multiple entities – and it allows for extending new features that can be shared with other developers.

# Chapter 3. Technical Implementation

Non-functional requirement: Have the web application run on the desktop and mobile versions of Chrome, Safari, and Firefox. How specific should I go?

## 3.1 Design of the System

For the system's design, I started with several sketches for what the webpage could look like. Due to this being my first foray into mobile development, sketching out some designs and jotting down my thoughts really helped me understand the space I had to work with.

Below was my first sketch of what the opening screen to the website could look like and, looking back, I can see the crowded nature of this design. Graphical user interface, diagram

Description automatically generated

## 3.2 Overview of Languages and Software Used

### 3.2.1 General Development Setup

For the development of this project, the text editor Visual Studio Code was chosen due to my familiarity with it and its flexibility. One extension for Visual Studio Code that was crucial to development was Live Server. This extension made it possible to have the website open while developing it. Any changes made were instantly reflected on the open webpage. Not only did this allow for the website to be run on my laptop but also by any device that is connected to the same local network such as my phone, which was critical for developing a website that is designed primarily for phones.

The main browser used during development was Chrome on MacOS. With its great dev tools and its excellent compatibility with all the latest technologies, it was a smart choice. However, due to its excellent compatibility, extra research had to be taken before using a specific feature in case it was not supported by other popular browsers. One such feature that caused me confusion was using ‘importScripts()’ from within a JS web worker to retrieve a file from a CDN. This feature is supported by Chrome, Firefox, and Safari Technology Preview (a beta version of Safari) but not by Safari despite the MDN Web Docs not mentioning it. It works with scripts that do not need to be fetched from outside the current folder. I am unable to find an answer to whether this is purposeful or a bug.

To test the implementation in real scenarios during development, the use of a website hosting service was needed. GitHub Pages filled my needs perfectly. As new features were added and pushed to my GitHub repository, the webpage would update automatically.

## 3.3 Discussion of OpenStreetMap Data Integration

### 3.3.1 Getting the Data and Displaying It

To access OSM’s rich database, the read-only API, Overpass API, was used. One can query this API to request specific data from a specific area. Another option would have been to pre-download the specific Lancaster area and send that over with the webpage. Choosing the former allows for my implementation to always stay up to date with the current database with little-to-no upkeep and to be lighter, to deliver the website quicker to the user’s device. However, this does incur a time penalty as there are load times that often vary drastically.

To combat this, a chunking system was implemented using the ‘cache’ interface in JS. After the map has initially loaded its first chunk around the user, the API is then queried eight more times (buildings and paths are queried separately) for each chunk that is at the edge of the current chunk. The responses are then stored in the local cache of the browser so that when the user travels close enough to the next chunk, the chunk can immediately load in front of them. Then, the next lot of queries are sent out for the next chunks. This keeps the loading times to a minimum as they navigate using the application.

Querying the API is executed in three requests; requesting the buildings is one request, roads and footpaths are another, and trees, rivers, bodies of water, and grassy areas are grouped in the final request. Doing it this way decreases the total time that the user will stare at a blank screen as they wait for the map to load. Additionally, if one of the queries is unsuccessful, the remaining two types of structures can still be loaded. If a query does fail, a new query with the same parameters is sent out. If it fails again, it will retry 8 more times with the time between each query doubling starting at 1 second.

Overpass Turbo, the interactive frontend to Overpass API, was used to construct the API queries required. The website allows you to enter and run an Overpass query and the results are then displayed on an interactive map or as raw text. This tool allowed me to build syntactically correct queries and ensure that the requested data was correct.

In an effort to minimise time spent blocking the UI thread, JS workers were implemented to execute the fetching of the OSM data. Blocking the UI thread should be kept to an absolute minimum as it heavily affects the responsiveness of the website. Each of the three queries has a worker that it can use and when a fetch request is needed, a message will be sent to the worker via a ‘postMessage(message)’ method call. The message sent to the worker will contain the query and the name of the cache where the response should be stored. The worker will then check the cache to see whether this request has already received a response. If it has, it sends a message back to the main thread with the response, otherwise, it sends the request to Overpass API. Once the worker has received the response from the API call, it will store the response in the cache and return the response to the main thread.

To parse the data from the API query, osmtogeojson.js was used. osmtogeojson.js is a JS library that does exactly as advertised; it takes the response from an OpenStreetMap API call and converts it to a GeoJSON. The reason for doing this is that it rearranges the data into an easier-to-use format. Figures 1, 2, and 3 illustrate what it does.



Figure 1- Raw text from Overpass query Part 1

Figure 2 - Raw text from Overpass query Part 2

Figure 3 - GeoJSON version of the data

Figure 1 and figure 2 show the data received from the response and figure 3 shows how the data is rearranged by replacing the node IDs with the nodes. It also figures out the type of geometry it is. In this case, it has been identified as a polygon.

Do I add an image of my code like above?

## 3.4 Explanation of 3D Rendering and Interactive Components



Figure 2 - Function from the implementation that is called to load buildings.

A-Frame is built to be simple to keep the entry requirements to develop for VR/AR low.

* Talk about how the map is rendered using A-Frame and Three.js.
* Talk about how the user can interact with the map on desktop and mobile.

# Chapter 4. User Study

For the user study, the participants were asked to walk to two places on campus. The first place they were instructed to walk to was Barker House Farm located in the Cartmel college area. The second place was Hawkshead which is a residential building in Furness college.

Every odd participant, i.e., participants one, three, and five, was instructed to utilise the application on the first walk to Barker House Farm and on the second walk to Hawkshead, they were instructed not to use it. This was swapped for every even participant, i.e., two, four, and six. For the walk not using the application, if they did not know where either Barker House Farm or Hawkshead is, that walk was skipped and the next walk or the interview was completed.

During both walks, the participants were asked to talk aloud about their experience and anything note-worthy was recorded using a pen and paper.

## 4.1 Description of Research Questions

Below are the questions asked and an explanation of why the question was chosen.

1. What year of study are you in?

This question is to help gauge the participant’s familiarity with the campus.

1. Did you find it easy to navigate the webpage?

This question was asked as an icebreaker conversation and to get a little feedback on the generic functionality of the webpage.

1. Was the 3D environment helpful in understanding where you were situated in the real world?

This question was asked to analyse whether the 3D nature of the implementation was effective in increasing the user’s situational awareness.

1. Was the 3D aspect helpful for navigating to the destination?

This question was asked to gain insight into whether the 3D environment was helpful in navigating.

1. How likely would you be to use a production-quality 3D map application over a traditional top-down view? (Never, rarely, sometimes, often, always)
2. What was your reasoning for that answer?
3. How often did you look at the top-down mini-map in the top-right corner? (Never, rarely, sometimes, often, always)
4. How was the performance on your device? Was it acceptable or too slow?
5. Is there anything that comes to mind that could improve the 3D environment?
6. Are there any improvements that could be made to the navigation system?
7. Are there any other features or improvements that come to mind to improve the overall design or usability of the website?
8. And finally, is there anything you noticed that you would like to ask about?

## 4.2 Explanation of Methodology for Data Collection

## 4.3 Analysis of User Feedback

## 4.4 Evaluation of 3D Map

## 4.5 Comparison of Map Features to Existing Solutions

# Chapter 5. Conclusion

## 5.1 Recap of Key Findings and Results

* Here, list the aims from the first chapter and talk about if they have been met.

## 5.2 Discussion of How the Project Could Be Extended

* Talk about the height map.
* Talk about how textures could be added for different structures such as stone vs concrete vs gravel paths.
* Maybe talk about scanned images of the buildings and paths that could be overlayed over the map.

## 5.3 Final Thoughts on This Project

End on a happy note ☺

References

|  |  |
| --- | --- |
| [1] | E. Andrews, “8 Remarkable Early Maps,” 22 August 2018. [Online]. Available: https://www.history.com/news/8-remarkable-early-maps. [Accessed 11 March 2023]. |
| [2] | Mobile Phone Museum, “Benefon ESC!,” [Online]. Available: https://www.mobilephonemuseum.com/phone-detail/esc. [Accessed 10 March 2023]. |
| [3] | “Navigation - Worldwide,” Statista, February 2023. [Online]. Available: https://www-statista-com.ezproxy.lancs.ac.uk/outlook/dmo/app/navigation/worldwide#revenue. [Accessed 11 March 2023]. |
| [4] | E. Reid, “A look back at 15 years of mapping the world,” 06 February 2020. [Online]. Available: https://blog.google/products/maps/look-back-15-years-mapping-world/. [Accessed 11 March 2023]. |
| [5] | D. Nowell, “£24m management school building now open at Lancaster University,” Lancashire Post, 21 May 2021. [Online]. Available: https://www.lep.co.uk/business/ps24m-management-school-building-now-open-at-lancaster-university-3245393. [Accessed 9 March 2023]. |
| [6] | “Most popular mobile apps in the United Kingdom (UK) in September 2022, by reach [Graph],” UKOM, September 2022. [Online]. Available: https://www-statista-com.ezproxy.lancs.ac.uk/statistics/1282098/top-mobile-apps-uk-by-reach/. [Accessed 12 March 2023]. |
| [7] | “Leading mapping apps in the United States in 2022, by downloads [Graph],” AppMagic, 07 February 2023. [Online]. Available: https://www-statista-com.ezproxy.lancs.ac.uk/statistics/865413/most-popular-us-mapping-apps-ranked-by-audience/. [Accessed 12 March 2023]. |
| [8] | Taginfo, “Database statistics,” [Online]. Available: https://taginfo.openstreetmap.org/reports/database\_statistics. [Accessed 27 February 2023]. |
| [9] | A-Frame, “Entity-Component-System,” [Online]. Available: https://aframe.io/docs/1.4.0/introduction/entity-component-system.html. [Accessed 11 March 2023]. |