BSc Final Year Project

Utilising HTML5 and Vector Maps to create an offline map application

Project Plan

Adil Mushtaq

Abstract

An overview of the aims and objectives for the project.

Online map applications have been an essential utility for a large majority of the population with users varying from cyclists to long distance HGV drivers. However, both users sometimes may travel through areas with limited or no internet connectivity or in the case of the latter travel through regions where they may need to pay for roaming charges to use the service.

Existing solutions to this problem include google maps' ability to download and store maps/directions as image tiles. It does this using raster maps which generates an image based of the location of each pixel[2]. The alternative to this is vector maps which uses information about connected vertices to generate the same image. Although raster maps are effective at storing map data and displaying it even when there is no internet connection, the large file size combined with the loss of directions and searching capabilities make it less than ideal in the circumstances mentioned above. This means that the smaller file size of vector maps as well as the ability to store Points of interests (POIs) as items within the map file means that they can be used without the same limitations as a raster map. However, the disadvantage of this technology is that it requires a higher level of client level hardware to process and display information[1]. Despite this, modern mobile devices are much more powerful than ever and so the impact of this problem may be negligible. OsmAnd is a notable example of how this concept can be applied and my project will attempt to replicate its successes in a web application context.

In this project, I will use HTML5, JS, CSS and OpenStreetMap (OSM) to develop a browser-based, vector maps application with offline capabilities. To do this, I will use the advanced features of HTML5 combined with JS and CSS to develop a usable interface for the application. I will initially have to revisit standard HTML and CSS to relearn the basics needed to create a simple web page before moving onto the features of HTML5 which will allow me to create canvas objects. This will allow me to process the vector data received and display roads, paths, and buildings. Recent updates in HTML5 also allow for further support for mobile browsers so that users can also use the app on the move in a lightweight package. The utility of HTML5 canvases would allow me to directly process and display the vector data required to generate the image seen by the user.

Alongside this, I will need to revisit JavaScript to develop the client-side scripts to allow users to interact with the map and, as an extension, search for POIs both online and offline. The data I will use for this will be gathered using the OSM database which will then be stored locally using an XML format. Another key aspect of the project is the ability to work offline which would rely on application caching. However, after further research and a conversation with my supervisor, it was made apparent that this was a deprecated method of doing such and so I should focus on using service workers which check for availability of resources in local storage and retrieves them when available. This would be in collaboration with an indexedDB which will store the data from OSM.

Finally, as an extension to the project I will attempt to develop a method for the user to dynamically download data from OSM when there is an internet connection available. This will be a toggleable feature which will automatically download data for tiles within a user specified radius of their current location when they are connected to the internet. This will provide a level of reliability for the user as they will always have local map information but also ensures that it doesn't overwhelm local storage.

Timeline

First and second term milestones, both reports and programs. planning and timescales.

To manage my time effectively, I decided to split the project into two phases which will correspond with term 1 and 2.

In term one, I will focus on researching the required technologies for the project with supporting reports and proof of concept programs. I will focus on three different concepts as a part of this which I will split into rough 2-week periods. The first would be creating a basic “hello world” offline HTML5 application to understand the core concepts, such as canvas objects, which allow for offline web application to developed. The next would be to use indexedDB to create a to-do list application which will allow me to understand how service workers and local storage are used. The last one will focus on using the data from OSM and work on how to process and display the data received. To end the term, I will collect all of this and start work on the interim report.

Term two will be when I consolidate everything I have learnt and start work on the actual final product using the techniques and tricks I learnt over first term. I will spend the first couple weeks starting work on the application’s user interface, creating a template for the webpage. Then I will focus on the back-end foundation which will gather and store information from OSM in an indexedDB. Finally, I will work on the scripting to incorporate the two segments of the application into a working product. Once I am happy with the quality of the product, and if time allows for it, I will also add the additional features mentioned in the abstract.

Term one

Week 1 (w/c 26/09/22) Research basis of project.
Week 2 (w/c 04/10/22) Research differences in mapping technologies. Submit project plan.
Week 3 (w/c 11/10/22) Refresh knowledge on HTML, JavaScript, CSS, and research how to utilise HTML5 and canvas objects to create a demo offline application.
Week 4 (w/c 18/10/22) Consolidate research from previous week in a report.
Week 5 (w/c 25/10/22) Research service workers, local storage and indexedDB. Start work on a to-do list application. Supervisor meeting.
Week 6 (w/c 31/10/22) Complete previous weeks application and consolidate research in a report.
Week 7 (w/c 07/11/22) Research how to process data received from OSM and possible data structures that would be compatible with format in which data is received.
Week 8 (w/c 14/11/22) Consolidate research from previous week in a report.
Week 9 (w/c 21/11/22) Start work on interim report and finish off any remaining research. Start presentation. Supervisor meeting.
Week 10 (w/c 29/11/22) Finish and submit interim report. Complete presentation.
Week 11 (w/c 05/12/22) Presentation.

Term two

Week 1 (w/c 09/01/23) Begin work on front-end design and create template for the application.
Week 2 (w/c 16/01/23) Continued.
Week 3 (w/c 23/01/23) Design settings page where user can adjust preferences within the application.

Week 4 (w/c 30/01/23) Start implementing methods of collecting and displaying collected data from the OSM DB.
Week 5 (w/c 06/02/23) Continued.
Week 6 (w/c 13/02/23) Start work on storing the data collected locally to be used offline.
Week 7 (w/c 20/02/23) Continued.
Week 8 (w/c 27/02/23) Test program under load and check for possible optimisations.
Week 9 (w/c 06/03/23) Incorporate search features and login system. Start implementation of dynamically downloading data when connected to the internet.
Week 10 (w/c 13/03/23) Continued. Final testing.
Week 11 (w/c 20/03/23) Submit final report.

Risk Assessments

Risks and mitigations.

Underestimation of task complexity

Although HTML, JS and CSS are technologies I have experience in, and I have a foundation in web development, offline web applications are something I have not worked with in the past so it may take longer than expected to research and understand these new concepts. Because of this, I have ensured that I have given myself enough time to research and understand each one. However, if there is a point at which time or ability becomes an obstacle to implementing a core technology or feature, I will adjust the scope of my project by removing additional features that are not compulsory based of the project specification to reduce the load and improve overall code quality. For example, the login system.

Incompatibilities between technologies

In this project, there are quite a few interconnected technologies that need to be compatible with each other for my final product to work. However, there may be a situation where these technologies may not support each other and thus I will have to find alternatives. This issue has already arisen in the case of application caches being deprecated and I had to speak to my supervisor to find the updated technology. This is something that I will keep in mind in my research this term.

Performance of final software

Vector maps are complicated and even though OSM has an easy format in which I can gather data from the database (XML) it will still take a significant amount of processing before it can be used to display information in a way the user can visualise and understand it. This means I will have to keep checking the ability of my product to deal with this without impacting performance.

Effects of data source uptime on program availability

Since this project relies on secondary open-source data, there may be points at which the source is unavailable which would lead to my product also not having full availability. This means I will have to ensure that I have a backup source in case of such a failure as well as if there is a discrepancy in the data available.

Bibliography

[1] Elias, M., 2022. *Raster vs Vector Map Tiles: What Is the Difference Between the Two Data Types?*. [online] maptiler. Available at: <https://documentation.maptiler.com/hc/en-

us/articles/4411234458385-Raster-vs-Vector-Map-Tiles-What-Is-the-Difference-Between-the-Two-Data-Types-#:~:text=Raster%20map%20tiles%20are%20larger,demanding%20on%20end%2Dusers%20hardware.> [Accessed 5 October 2022].

[2] Google Developers. 2022. *Vector Maps | Maps JavaScript API | Google Developers*. [online] Available at: <https://developers.google.com/maps/documentation/javascript/vector-map#:~:text=The%20raster%20map%20is%20loaded,served%20to%20your%20web%20app.> [Accessed 5 October 2022].

[3] Pandey, D., 2022. *Difference Between HTML and HTML5 - Scaler Topics*. [online] Scaler Topics. Available at: <https://www.scaler.com/topics/difference-between-html-and-html5/> [Accessed 5 October 2022].

[4] HTML5 - MDN Web Docs glossary: definitions of web-related terms: MDN (no date) MDN Web Docs Glossary: Definitions of Web-related terms | MDN. Available at: https://developer.mozilla.org/en-US/docs/Glossary/HTML5 (Accessed: October 6, 2022).

[5] (no date) OpenStreetMap. Available at: https://www.openstreetmap.org/about (Accessed: October 6, 2022).

[6] XML ESSENTIALS (no date) W3C. Available at: https://www.w3.org/standards/xml/core (Accessed: October 6, 2022).