GPS Treasure Hunt

Final Project Dissertation

Oliver Barnes
1905121

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

May 3, 2022

Abstract

In a post-COVID world, exercise and socializing are returning to the front of the public conscience. People are looking for ways to get back outside, get healthy, and meet other people again. Mobile computing is as ubiquitous as ever, and there are a lot of technological solutions offered by mobile platforms to help motivate people to meet others and get fit.

The solution I have offered here is a GPS treasure hunt app for the Android platform. The idea is that a user should be able to bury a virtual “treasure” by walking in a pattern outside, and another user should be able to follow that same pattern in order to access the treasure that has been buried. I have achieved this by using the location capabilities of the Android system, the Firebase platform to store persistent, centralized data, and the Google Maps SDK to create a helpful interface for the user.

The idea is that this gamified form of exercise – a system that offers a reward for walking around in the real world – could help ease this problem of a sedentary and isolated post-lockdown population. In this document I will discuss how I built my solution, how effective it was at achieving its purpose, and its place in the market when compared to other apps that offer a similar experience.

Table of Contents

Chapter 1

Introduction

Throughout 2020 and 2021, the UK has seen varying degrees of lockdown measures taken to limit the spread and impact of the COVID-19 pandemic. Most of the rules and restrictions implemented were based around the ideas of keeping people in their homes and away from other people [1].

Work-from-home orders, curfews, and social distancing were among the measures taken, and they did not come without side-effects. A lot of people were exercising less, eating more, and were less motivated to control their diets [2]. In an already sedentary population where obesity is rising at unprecedented levels [3], this could be very dangerous, and is overall negative for public health.

A novel suggestion to counter this crisis is the idea of gamifying physical activity, and it has already shown some progress in this regard [4]. Some users of the mobile app *Pokémon Go*, an augmented-reality GPS-based game, reported an overall increase in physical activity as a direct result of using the app.

The idea for this project was to create a mobile app in which users partake in a "GPS treasure hunt". A player can hide virtual treasure in the real world by walking a route that they create, composed of many "waypoints" that would define a given path. Other players can then follow the path they created in order to collect this virtual treasure.

1.1 Motivations

As already stated, creating a product that provides a gamified form of exercise was part of the motivation for this project, especially after the events of the COVID-19 lockdowns. The use of mobile devices is often characterized by physical inactivity, but this project's aim was to combine both exercise and the use of a mobile phone.

Another motivation is to create a more social post-COVID climate. Lockdown severely limited the public's ability to meet new people, or participate in activities with people they already knew, and as such public mental health suffered [5]. An app like this could help push some people out into their local communities to partake in and curate these digital "treasure hunts".

On a more personal level, this project helped me develop my skills. I had never developed anything on a mobile platform before, so this broadened my skillset and provide me with an opportunity to showcase what I am capable of.

1.2 Aims

The aims of this project were as follows:

- To create an Android-based mobile app with GPS functionality. This app should:
    - Allow users to create a route in the app with their movement in the real world. At the end of a given route they should be able to choose what data they want stored in this location.
    - Allow users to follow a given route and retrieve the data that someone else has stored there.
- To create a server script that responds to requests from the app for data those other users have entered. This will allow users on different devices to collect the same data by following the same route.

Chapter 2

Background Research

2.1 Related Work

2.1.1 Gamification-centric

Pokémon Go

Pokémon Go is a product released by Niantic in 2016. Based on the popular *Pokémon* IP, it is a mobile game in which the aim is to collect a variety of fictional creatures through completing mini games within the app, as well as visiting real-world locations. The app works heavily off location-based elements, with the creatures you encounter being related to the environment you are in, for example fish-style creatures are usually found close to bodies of water. Some real-life landmarks are signified by markers (or *Poké-stops*) within the app, providing helpful in-game resources when visited.

Despite being a GPS-based game, it's not really fitness focused, with most of the emphasis being on the mini-game and social aspects. The social element of the game is interesting, allowing for more co-operative as well as competitive play.

Figure 2.1: The game's map view or "overworld" through which all other elements can be reached.

Figure 2.2: The "catching" mini game, in which the player must throw a ball in order to tame a creature.

This product is one of the best known in the novelty GPS app space, with 14,000,000 downloads on Android alone (as of 10/2021) [6].

Pokémon Go is a lot larger and more complicated than what I'd hoped to implement but gave a good starting point for thinking about how location-based interactions could work, what a good interface can look like, and how to keep a user interested and motivated to use an app like this.

Actionbound

Actionbound [7] is an app for playing interactive scavenger hunts. Users create their own treasure hunts (these can use text or image-based clues, GPS elements, or QR codes) from their browser, to be played on a mobile device with the app installed. There's a very open-ended nature to the platform, as it can be used for anything from corporate team building activities, to guided tours, to birthday scavenger hunts.

As such the motivation behind this platform appears to be to put as much freedom in the hands of the user as possible, as opposed to having a strict focus on fitness, socializing, or monetization (especially obvious in the case of *Pokémon Go*).

Out of all the related work I looked at *Actionbound* offers the closest user experience to what I wanted to build. It's much larger in scope than anything I was hoping to build but certain parts of it, especially the elements of it that use location interaction, influenced my thinking when it came to making my product. It also made me question the degree of freedom I wanted my app to offer to users, given that *Actionbound* allows users to upload their own photos or video.

Enriching Tourist UX via a Location Based AR Treasure Hunt Game

This study [8] shows not only how location interaction can be used with a mobile app, but also how AR elements can be added for a more cohesive experience. A mix of 3D models, photos, video and interactive quiz elements are employed to teach the user about certain real-life locations, offering a kind of digital guided tour experience. The use of the location driven UI and game elements showed promise as far as helping users to learn in an interactive and fun way.

While this project's focus is more on creating an educational experience, its use of gamification and location elements are close to what I wanted to achieve.

2.1.2 Exercise-centric

“Working out for likes”: An empirical study on social influence in exercise gamification

This study talks about how important a social aspect can be to a gamified exercise platform. Their ideas were centred around how behaviours in individuals are affected by those around them, and how that could apply to something like a mobile app that measures distance walked [9]. Their results show that there is a social impact, and that it increases with the number of others they know also using the platform.

The points put forward by this study were important to consider for my project, given how much I’d talked about a gamified approach. It showed how competition can be leveraged to motivate users, and what impacts a social element can make to this kind of app. It did inform some of my thinking when it came to making the final product

Is Gamification Effective in Motivating Exercise?

This study [10] is about how the usage of an app that gamifies exercise can motivate someone to increase the amount of physical activity they engage in. The app in question is called Fitocracy [11], which is an online social platform where users can log exercise activity in order to gain points. The platform also has leaderboards where users can see where they rank amongst their friends by their point totals.

It was found that, generally, users of the platform were much more motivated to exercise but that it was the social features that were essential to this motivation, not just the gamified parts. Users were found to use the social elements to encourage each other, as well as competing. The study also mentions that some users were less motivated by the platform, as they disliked the competitive nature of it.

This raised a lot of similar points to the “Working out for likes” study, in that a social element can be helpful, and that competition is a good motivator. One thing that this platform offered (that I have no plans to attempt) was exercise tailored to the user based on what they wanted to achieve, which I think also plays a large part in keeping user interest.

2.2 Tools & Techniques Used

2.2.1 Google Maps API

The Google Maps API allows developers to use elements of Google Maps in their own applications. More specifically, it allows for the display of an actual map to the user, as well as custom map markers, and the location of the user on the map.

I used the Google Maps SDK for Android. This allowed me to create the interface I envisioned for the product, one that would show the user their location on a world map, along with important information like locations of route markers. This was invaluable to my project, as I used the map interface provided as the main method of interaction with the app.

2.2.2 Fused Location Provider

The Fused Location Provider is a part of the Android system that manages the location technology on board the device. It simplifies the use of location data within apps by handling everything to do with GPS and Wi-Fi signals, and simply providing accurate location data when queried.

Usage of this by the app was necessary since the whole concept is so heavily dependent on location data. It's used for everything from checking the user's current location against nodes on a route they're following, to recording user's location when they place a route node.

2.2.3 Firebase

Firebase is a platform for mobile and web app development. It allows for user authentication within apps as well as storing data in real-time databases. It greatly simplifies the way in which developers can achieve these things with their apps, as it means that no server-side code has to be created by a developer. They just need to know what is stored within their Firebase instance, and program their application client based on that.

Firebase was used in my project for user authentication and storing persistent data, about the users and the app. User inventory data, user scores, and route data are all stored in a real-time database on the Firebase platform.

Chapter 3

Implementation

4.1 Problem Analysis

When starting to break down the problem, I looked over my aims, I realized that the app was formed from two main interactions, and they were connected by an over-arching interface. I made flow diagrams to describe how these interactions and the app navigation would work.

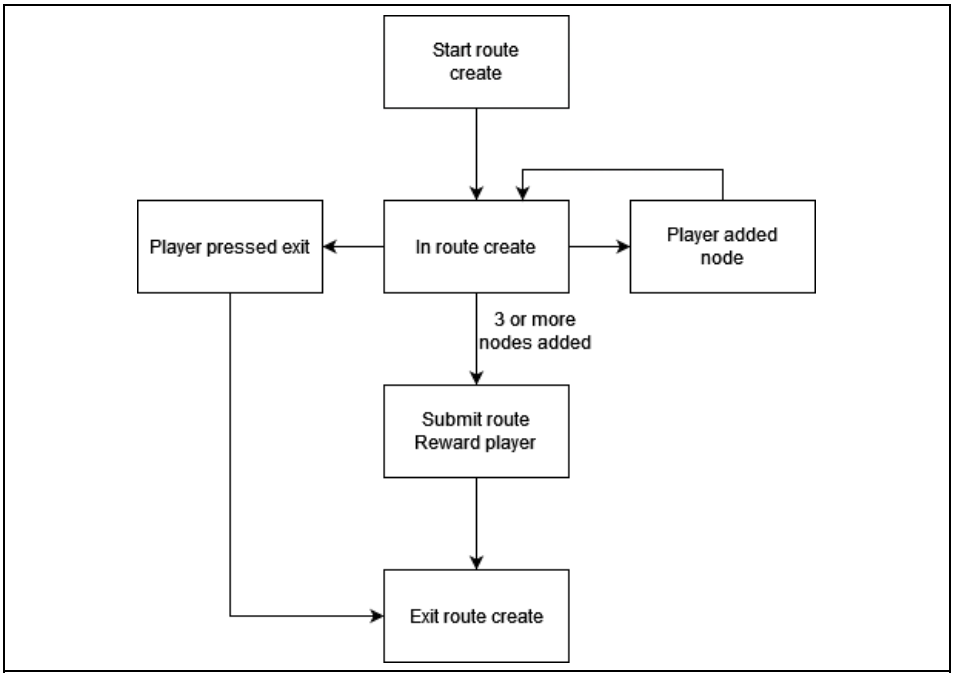


Figure 4.1: Route creation flow chart.

Starting with the first of my aims, I decided how I wanted the route creation interaction to work. The user would initiate the interaction, and from they would have three possible choices, to exit the interaction, to add a node to the route, or submit the route as being complete. For the route to be submitted, first three nodes must be added to it. I decided to impose this restriction so that users would likely consider making routes that are more interesting than short, straight lines. Upon submission of a route, the player is rewarded, and they are then redirected out of the route creation event.

The submitted route, much like the player's reward were to be stored on a central server. The idea of rewards came from how I wanted to gamify the experience offered by the app, that the player should receive something for their activity.

I decided to alter the initial aim to the extent that the user shouldn't be allowed to decide what would be given from following their route. This was because I was worried about what users might do when given the freedom to store their own photos, video, or even text on a service with no planned moderation. I also felt that allowing users to store whatever they want didn't help as far as gamification, and that if the app decides what is provided then a more cohesive experience is offered. It guarantees a consistent reward for completing a task.

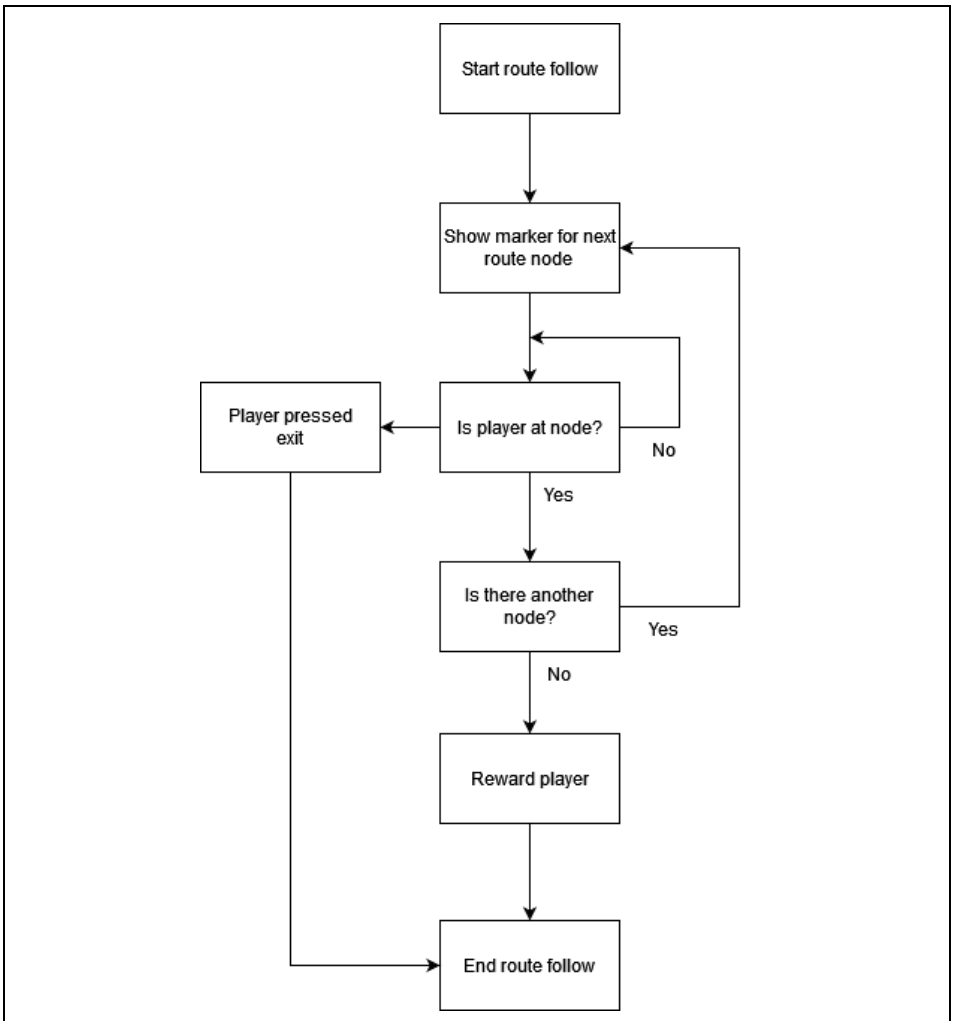


Figure 4.2: Route follow flow chart.

The next aim was to allow users to follow routes that other players had created within the world. This interaction would work similarly to the last, the user would initiate the event, and from the moment the first route marker (showing the location of the first node) was displayed, they would have the option to exit the interaction. On completion of the route, the user would receive their reward, and the interaction would finish.

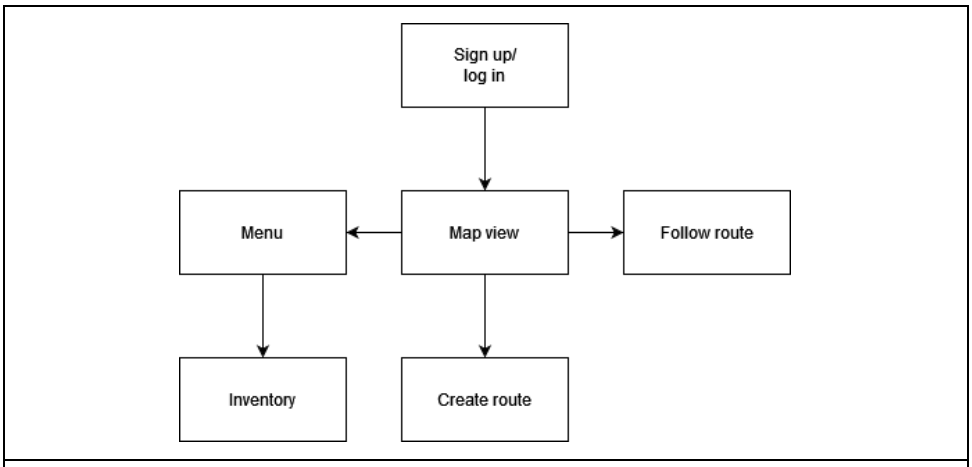


Figure 4.3: Navigation flow chart.

Figure 4.3 shows how a user would navigate around the app. They would begin on a log-in page, and upon creating an account or logging in to an existing one, they would then be moved to the map view. From this map view (which would show the user's location on a world map), the user would be able to follow an existing route, create a new route, or navigate to the menu screen. After navigating to the menu screen, the user can move to their inventory, where they would be able to view their rewards.

4.2 Development

I started the development stage by drawing concepts for how the interface would look and creating the XML versions in Android Studio. Apps created for Android are composed of "activities", effectively the different screens or layouts that the user will see. I knew my app was going to have 5 of these activities, the main map view, the signup and login pages, the menu, and the inventory view.

The login and signup pages were created quickly using Firebase for authentication. They were the simplest to implement, as both had roughly the same visual layout, and similar code that made them function. Both send data to Firebase, to check if the user has entered correct login credentials, and to create an entry for a new user respectively.

Next was the menu activity, which was simple too, only composed of some buttons and text. The inventory activity was accessible from here, and that was more interesting to implement, as it involved displaying data from Firebase.

On Firebase I used a JSON database, where data is separated into name/value pairs. I planned to have it built from three distinct sections. One for user inventory data, another for route data, and one for data about the user.

The inventory section grouped items by the user that owned them, each item had its own UUID (a randomly generated value that should be unique), and individual item data was stored under that (information like the item name and the quantity that user owned).

The inventory activity would send a request to the database to get inventory data for the user currently logged in, and then display that data in a RecyclerView (a scrollable list). This data would be passed to an adapter class, a class designed to map the received data onto an XML object to be displayed in the list.

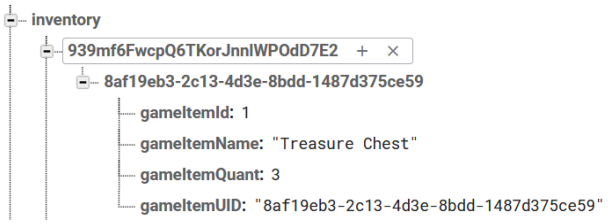


Figure 4.4: Inventory database example.

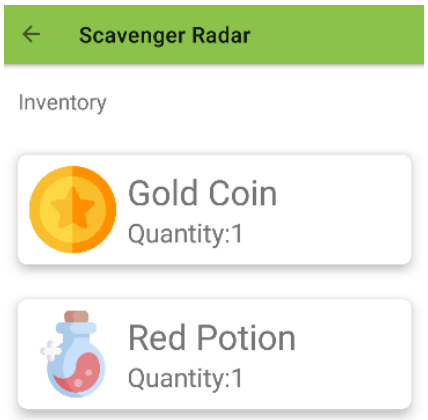


Figure 4.5: Inventory view.

The map activity was the most challenging to implement as it contained the bulk of the functionality. I started with a map activity template provided as part of Android Studio and the first things I implemented were the button UI elements. One would open the menu screen, another would centre the map on the user's current GPS location (using a function built into the Google Map fragment on the template), and the last one was intended to initiate the route creation event.

I had decided that in order to create a route for others to follow, a player must have a specific item (a "treasure chest") in their inventory. Upon submission of a created route, this item would then be removed. The idea behind this was that I didn't want the user to be able to create an unlimited number of routes in a small space that would fill up the map with markers. They would have to follow other people's routes in order to have a chance to create more routes of their own.

I set up a call to the inventory part of the database to request the number of "treasure chests" the user had, in order to display this on the route creation button. This would mean that they would know how many routes they could still create without entering their inventory. In the event they did not possess this item, the button would not be visible to them.

The next part was creating a function that would populate the map with markers for routes that the user would be able to follow. The user could choose which route to follow by selecting the marker for it, and these markers would be at the location of the first node of their given routes. This involved another call to the database, this time to the route data section.

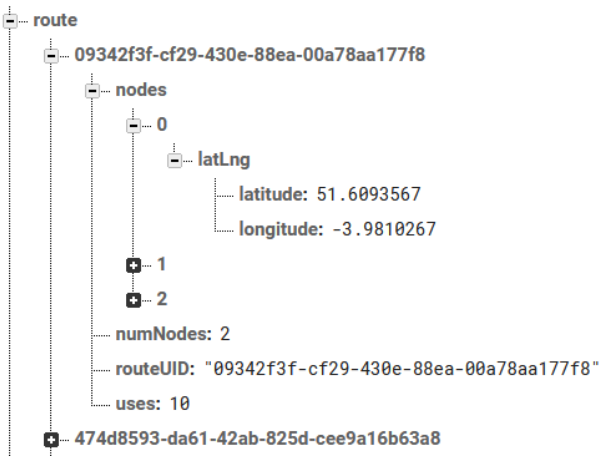


Figure 4.6: Database route data structure.

Every route has a UUID, under which all data about the route is stored, including the number of nodes on the route, and the latitude and longitude of all the route nodes.

Next was the implementation of the route creation interaction. Since this interaction was part of this activity, I had to make it clear from the UI that the user had started creating a route. I did this by hiding some UI elements (the button that starts the interaction, the menu button) and adding the required ones for the interaction in. I also thought it would be helpful to clear the map of all markers, so the user would not be confused by them.

The three possible actions in this interaction are adding a node, submitting the route, and exiting the interaction, so there is a button for each. On pressing the button to add a node, a check is carried out to see if the user is within 10 meters of their last node placed. If they are, an error is displayed. If not, their current location is added to an ArrayList of nodes, and a marker is placed at their location to signal the position of the node. Once the user submits the route (only possible if there are three or more elements in the ArrayList), it is sent to the database, they receive their reward, and a marker for their route is added to the map for all users. The UI returns to normal on exit from the route creation interaction.

In most of the app, locations are represented using the LatLng data type, which just contains latitude and longitude values. This was helpful for a few reasons, most of all that the native “Location” type is fairly difficult to use, but also that LatLng is compatible with the LatLngBounds “contains” method. This method checks if a

given LatLng exists within a set of LatLngBounds. This was helpful as far as checking that nodes are not placed at the same location, and in the final part of the map activity – the route follow interaction.

The route follow interaction was by far the most difficult part of the app to implement. In order to access it, a user must click one of the markers, and then confirm that they would like to follow that route. Once again, the UI changes significantly, this time the only buttons that remain are one that allows the user to exit the interaction, and the one to centre on the user's location. All map markers are removed, except for the one the user clicked on to start the interaction, as this is the location that the user is expected to move to.

Upon reaching the marker location, it will disappear, and a new marker will appear at the next node on the route. This will repeat until the user reaches the end of the route, at which point they will receive their rewards, and the interaction will end. The map screen will then return to normal, with the menu button reappearing, and the map repopulating with route markers.

The reason this was so difficult to implement is due to how location updates work with the location provider. Previously, it was possible to create location listeners within methods, but that has since been deprecated. I was able to achieve this by putting a check within a LocationCallback that would see if the user was within 10 metres of a node marker. Since it's within the LocationCallback, the check is run every time the user's location updates.

There are other ways of achieving this with Fused Location Provider, namely geofences, but I found them extremely difficult to use.

4.3 Finished Product

4.3.1 Screenshots

After some amount of testing and some minor fixes, I had a finished product that mostly adhered to the initial aims. Figures 4.7 - 4.14 are screenshots of the app.

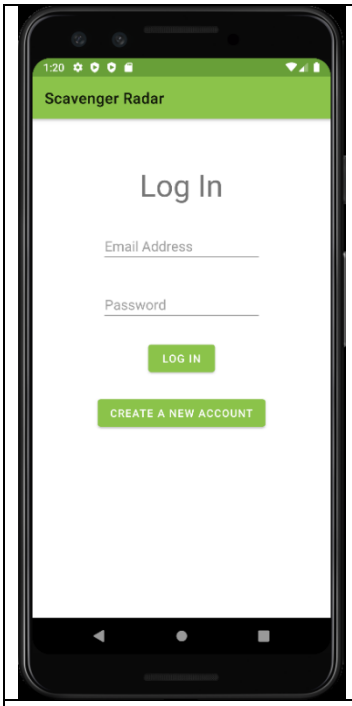


Figure 4.7: Login screen.

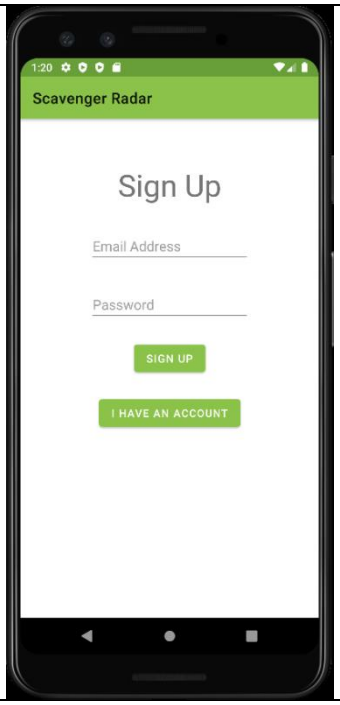


Figure 4.8: Signup screen.

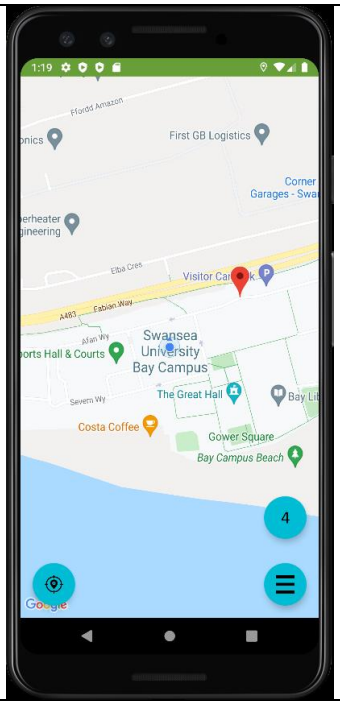


Figure 4.9: Map view.

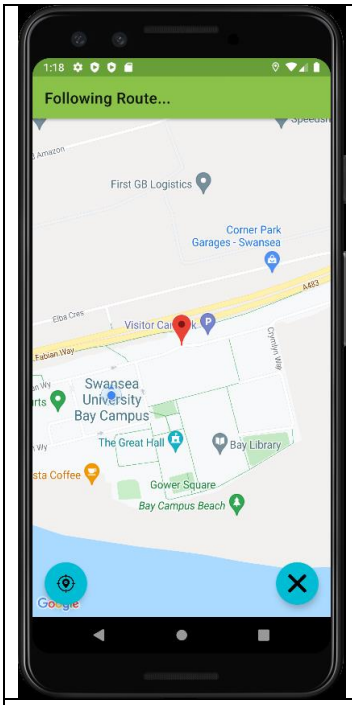


Figure 4.10: Route follow event.

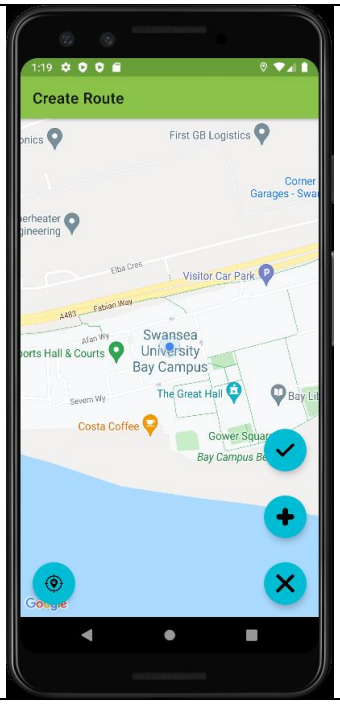


Figure 4.11: Route creation event.

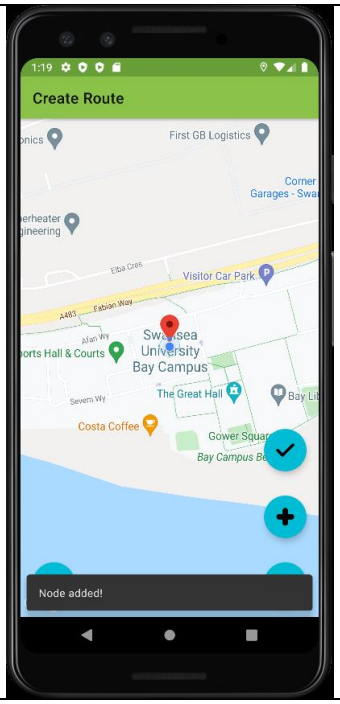


Figure 4.12: Route creation, node added.

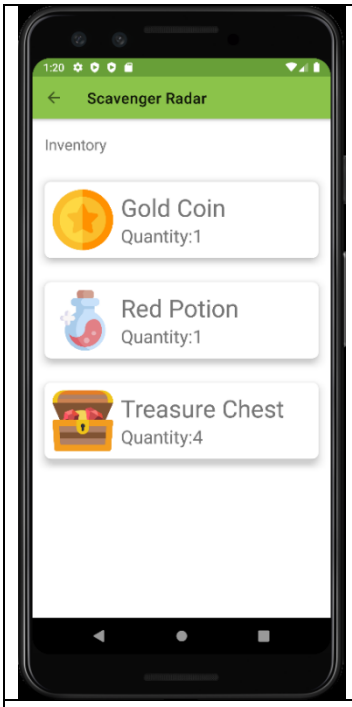


Figure 4.13: Inventory screen.

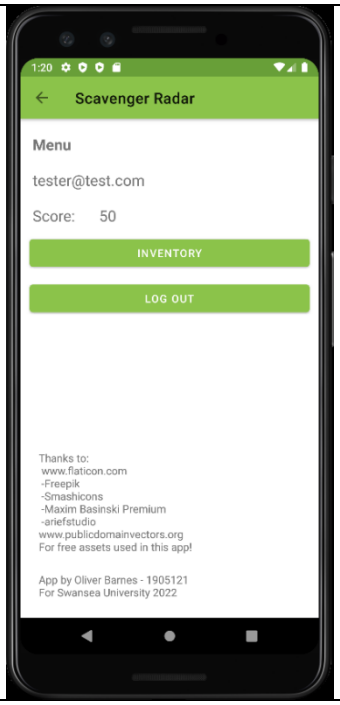


Figure 4.14: Menu screen.

4.3.2 Using the App

In order to use the app, the user must first either register or log in to a pre-existing account. They will then be presented with the map screen shown in figure 4.9. From here they can access all other functions of the app.

Creating a route

To create a route the user must press the button with a number on it, located above the menu button. To be able to see this button the user must have at least one treasure chest item in their inventory. These are required for creating routes and one is consumed once the user submits the route.

On pressing this button, the route creation event as shown in figure 4.11 will begin. From here the user can add nodes to a route. These nodes are the locations that a user must walk to in order to complete the route and collect treasure. Nodes are added by pressing the “plus” button in between the submission and exit buttons. A user cannot create routes remotely, they must be standing in the location in which they wish to place the node. A node cannot be within 10 meters of the previous node placed.

A route must be composed of at least three nodes, and once they have been placed the user will have the option to add more nodes or submit the route. On submitting the route, the user will be rewarded with treasure, and there will be a marker on the map screen for the route, which all users of the app will be able to see.

Following a route

The markers presented to the user on the map screen represent routes created by other users. Each marker is shown in the location of the first node of its respective route. If a user were to click on one of these markers, they would be presented with a dialog asking if they would like to begin following the route, and if they select yes, the route follow interaction begins.

From here, the only marker displayed on the map is at the location of the next node on the route. Once the user is within 10 meters of the node, the marker will disappear, and appear at the location of the next node after that.

When the user has visited every node on the route, they are rewarded, the interaction ends, and they are directed back to the map screen.

Chapter 4

Results and Conclusion

5.1 Evaluation of Finished Product

The project met its aims, runs well on a hardware device, and I think looks good visually. I think it is too shallow from a features standpoint though, there's not much to do in the app except collect treasure. Personally – I wouldn't use my app, but this isn't because I think it's a bad quality product, just conceptually flawed.

I created a short video demonstrating the app and put out a survey asking prospective users what they thought about it (see Appendix A for full results). I started by asking if they understood how to use it, and if the interface made sense to them, then moved to asking about whether they would use the app, and whether it fits its purpose.

The response as far as the UI was generally good – most thought they would be able to use the app based on the short demo, and that the layout made enough sense. As far as interest in the product, the largest response to whether they would download the app was "maybe", with only about 20% saying that they would download it. Feedback from one user said that the app "looks boring". Users suggested ideas like mini games, so that there was more purpose and fun to app interactions.

Only about 30% of users thought that the app would be useful as far as keeping fit, but roughly 50% thought that it could be good for socializing. This isn't great, given that these were the motivations for the project – but I think that it does signal a key issue with the final product.

If given more time, and the direction I have now, I would improve the app by adding social elements. I did consider this throughout development, wanting to incorporate a leaderboard using Google Games Services [12]. I still think that there would need to be a bigger social aspect than just a leaderboard to give users a reason to use the app, but I think it would be a good start.

5.1.1 Comparison to Related Work

What I've built here is a lot simpler than everything I looked as far as related work. I think that in a way, it's one of my product's strengths. It certainly seems the easiest to use when compared to other products, but this fact doesn't act like a real draw to it in the way that features of other products do. Pokémon Go's strengths are its brand appeal and its complexity. The brand pulls users in, and the complexity keeps them using the app.

I think the lack of a social aspect is the biggest detriment to the experience offered by my app. Two of the studies I looked at showed that while an app like this can be helpful in motivating a user to exercise, but this is because of the heavy incorporation of social elements.

I think that the gamification aspect offered by my app is quite weak too. The study where a location app is used for education leans heavily on its mini game aspects to be effective, the idea that you would go to a place and complete an action works better at keeping the user engaged than simply going to a place and being rewarded.

Compared to all the work I looked at, I think that my app needed to take greater influence from existing work, instead of just trying to be a simpler version of what already exists. I don't think that what I have built is very helpful when the market is full of properties that offer a more interesting experience, and the academic space is answering more interesting questions than if an app that uses location interaction can help people to get out more.

5.2 Conclusion

Overall, I think this project has been very interesting. Both as far as the actual development and implementation, but also the response to it. While I think that the right app could be effective at motivating people to exercise and socialize, what I have built isn't very good at these things. From the work I looked at I could see that social and gamified elements are great for keeping a user engaged – and I simply didn't make them a big enough part of the experience offered by my app.

What I've found from my project is why users will download and engage with apps in the first place, and that two big factors are social pressures and fun. If an app simply isn't fun, and there's no social gain from their participation, a user won't think that it's worth their time and therefore won't want to use it.

Bibliography

[1] Baker, Barber, Brown, Kirk-Wade. (2021). *Coronavirus: A history of English lockdown laws*. Available: https://commonslibrary.parliament.uk/research-briefings/cbp-9068/.

[2] Robinson, Eric et al. *Obesity, eating behavior and physical activity during COVID-19 lockdown: A study of UK adults*. Appetite vol. 156 (2021): 104853, doi:10.1016/j.appet.2020.104853

[3] Agha, Maliha, and Riaz Agha. *The rising prevalence of obesity: part A: impact on public health.* International journal of surgery. Oncology vol. 2,7 (2017): e17. doi:10.1097/IJ9.0000000000000017

[4] A.G. LeBlanc, J.P. Chaput. *Pokémon Go: A game changer for the physical inactivity crisis?* Preventive Medicine, 101 (2017), pp. 235-237, doi:10.1016/j.ypmed.2016.11.012

[5] Rhian Murphy and Chris Shine. (2021). *Coronavirus and the social impacts on behaviours during different lockdown periods, Great Britain: up to February 2021*. Available: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandwellbeing/articles/coronavirusandthesocialimpactsonbehavioursduringdifferentlockdownperiodsgreatbritain/uptofebruary2021.

[6] Pokemon Go, Google Play listing (2021). Available: https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=en&gl=us.

[7] Actionbound Website (2021). Available: https://en.actionbound.com/.

[8] M. Cauchi and D. Scerri, *Enriching Tourist UX via a Location Based AR Treasure Hunt Game*. 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin*)* (2019), pp. 199-204, doi:10.1109/ICCE-Berlin47944.2019.8966141.

[9] Hamari, & Koivisto. *"Working out for likes": an empirical study on social influence in exercise gamification*. Computers in Human Behaviour, 50 (2015), pp. 333-347, doi:10.1016/j.chb.2015.04.018.

[10] Goh, D.HL., Razikin, K. (2015). *Is Gamification Effective in Motivating Exercise?* Human-Computer Interaction: Interaction Technologies (2015), doi:10.1007/978-3-319-20916-6_56.

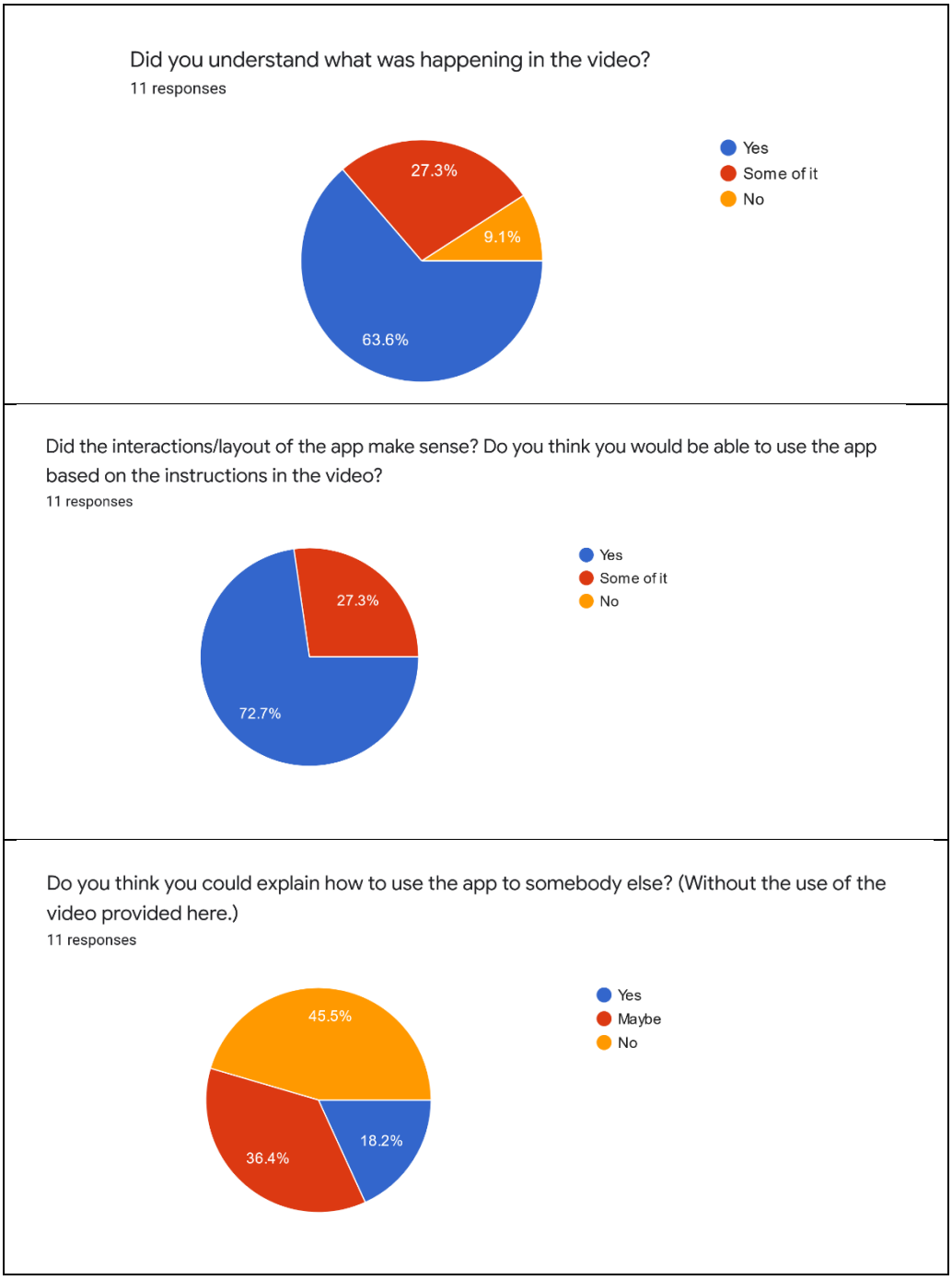
[11] Fitocracy Website. (2014). Available: https://www.fitocracy.com/.

[12] Google Play Games Services (2022). Available: https://developers.google.com/games/services/.

Appendix A

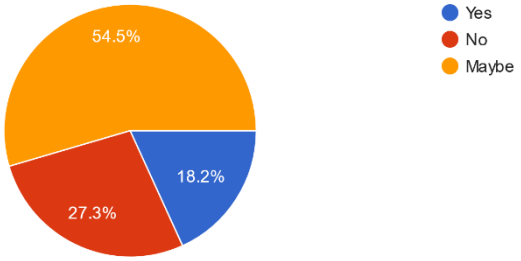
Results of App Survey

As mentioned in chapter 5, I put out a survey to find out what people's thoughts on my app were after they watched a short video on how it works.



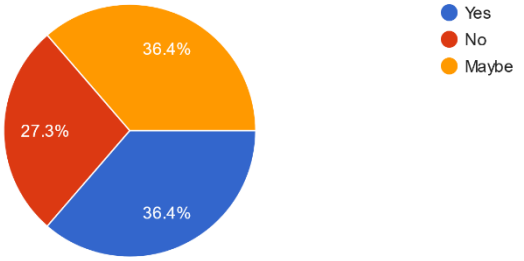
If this were a publicly available app, would you download it?

11 responses



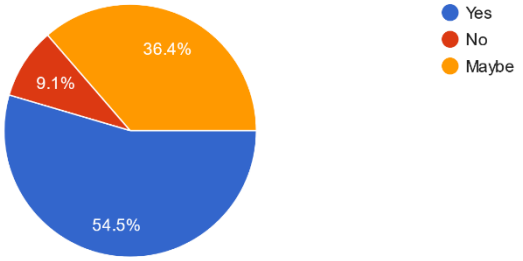
Do you think an app like this could be used to help people keep fit?

11 responses



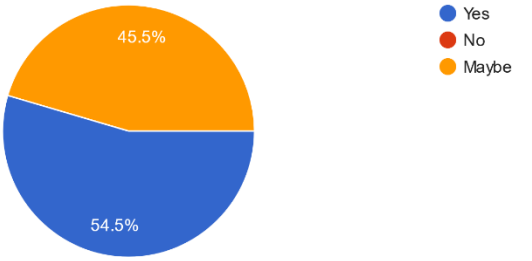
Do you think an app like this could help people to socialize?

11 responses



Do you think an app like this could increase interest in public spaces?

11 responses



What do you like about the app?

1 response

its simplicity

Do you think anything could be added to it to improve it?

2 responses

mini-games, something to collect

more interactions - things to do after you walk to places

Are there any parts of it you think are unnecessary?

1 response

the items

Why would/wouldn't you use it?

1 response

looks boring