Using GPS to create 3D Models of terrain in a sports mobile application

By

Jaco van Cranenburgh

Project unit: PJE40
Supervisor: Jonathan Crellin

April 2016

Abstract

Sports applications make use of GPS technology to help the user track their position, performance and speed. This is a technology that has proven to be extremely useful. However, these applications are usually limited to 2D representations of position – by displaying location on a flat map. This paper outlines the research, design and implementation of a mobile app that can display GPS location and altitude in 3D space. An artefact will be created that can track GPS location data and generate 3D terrain. This paper will then evaluate this artefact and the implementation process and discuss the applications viability.

Table of Contents

1 - Introduction

1.1 - Problem Overviews

There are many different sports applications currently available that make use of GPS to display the user's position and path. This is very useful as it allows the user to track their location and derive key performance statistics, such as distance, top speed, average speed and therefore aim to improve their overall performance. However, these applications are dependent on third party *maps* to project this GPS data onto. As well as this, most of these maps are 2D, 'flat' maps, lacking a display of altitude. Whilst this is adequate for most uses such as running or cycling on flat country, it leaves room for improvement for activities such as skiing, snowboarding, hiking and even running and cycling in mountainous terrain. In the mountains, altitude becomes a highly relevant metric for monitoring ones' performance. Furthermore, it is usually much harder to get access to a cellular or Wi-Fi connection in mountainous terrain, and third-party maps are often not available or very expensive to purchase.

The goal of this project is to research and implement a mobile application that can generate a 3D model of terrain from GPS data collected during activity. It will have no reliance on cellular or any other connection, or third party geographical maps and will only use GPS data

1.2 - Aims and Objectives

The aims of this project are to:

Conduct background research into GPS and terrain generation

- A literature review into relevant technologies and current academia will be conducted, that will help design and implement this mobile application.

Conduct a consumer research survey (primary research) to determine the requirements as dictated by potential end users

- A survey of this kind will help determine what features and benefits are most important in current sports applications, and these results can be applied to this project.

Produce a software artefact that can track GPS data, and generate a 3D model based on this

- The literature review and research survey results will be used to support the design and implementation of the software artefact itself.

Discuss further applications and potential future changes to the software

- Upon completion of the software artefact, there will be discussion of further research and work that could be considered and/or undertaken to improve the mobile application

1.3 - Development Methodologies

This project will be following the Waterfall model, including the stages described in this model, that is, "Requirements, Implementation, Verification and Deployment" (Powell-Morse, A. 2016). The Waterfall model requires that no previous step is revisited until the end ("Agile vs Waterfall" n.d.). In other words, you can only travel 'downstream' before returning to the top. This premise will help keep the project moving along and ensure the tight deadline of the 27th April 2018 is met, while also ensuring that the initial design is precisely followed throughout the project.

Another contender was the Agile Development process. The reason for not choosing this model was due to the fact it is designed to be used by teams, or 'scrums'. The goal of Agile is to be adaptive to changing requirements and working on different sections of a project in parallel. It allows rapid, continuous delivery of software (“What is Agile model” n.d.). Agile was not chosen due to these benefits not being deemed necessary, and as there is only one developer, a model like Agile is considered to add too much complexity to the development process.

2 – Literature Review

2.1 - GPS Formats

NMEA data (National Marine Electronics Association data) is a standard data format, supported by all GPS manufacturers which existed long before GPS was invented. It was originally created as the only uniform interface standard for data exchange between marine electronics in the 80's (HISTORY OF NMEA, n.d.), but has since become the go-to data format for GPS data across all industries. There are many different types of NMEA 'messages', which represent different GPS data and capabilities. For example, the following message in figure 2.1:

Figure 2.1 – Example NMEA message ($GPGGA)

```
$GPGGA,181908.00,3404.7041778,N,07044.3966270,
W,4,13,1.00,495.144,M,29.200,M,0.10,0000*40
```

This message represents a $GPGGA GPS position, using latitude and longitude. It then also denotes the Quality Indicator, number of satellites used in the coordinate, and other useful information such as altitude data. This is just one example format of a NMEA message, another example being $GPVTG, which shows speed over ground and tracking offset (Gakstatter, E. 2015).

Due to NMEA being the standard GPS format, receiving NMEA position in a mobile application is straightforward. On Android devices there is a driver that, once created, can receive the location of the device in various NMEA formats (GPS, n.d.).

2.2 - Types of 3D Digital Terrain Representations

Generating a 3D model from GPS requires first transforming the GPS data to a Digital Elevation Model (DEM). There are various types of DEM's, one of which is raster data in the form of height maps. Height maps are used to store elevation and positional location, which can then be used to generate a 3D model.

2.2.1 - Height Maps

"3D terrain using a heightmap" (2014) explains that a height map is usually a black and white 'print' of a landscape. This is also known as a raster image and represents the altitude of terrain at a given position in the image. The altitude is represented by the HSV (hue saturation value) at that pixel. This 2D image is then capable of being converted into a 2D array of heights, which can be interpreted by 3D modelling software to create 3D terrain.

Figure 1.1 – Height map of the earths' surface

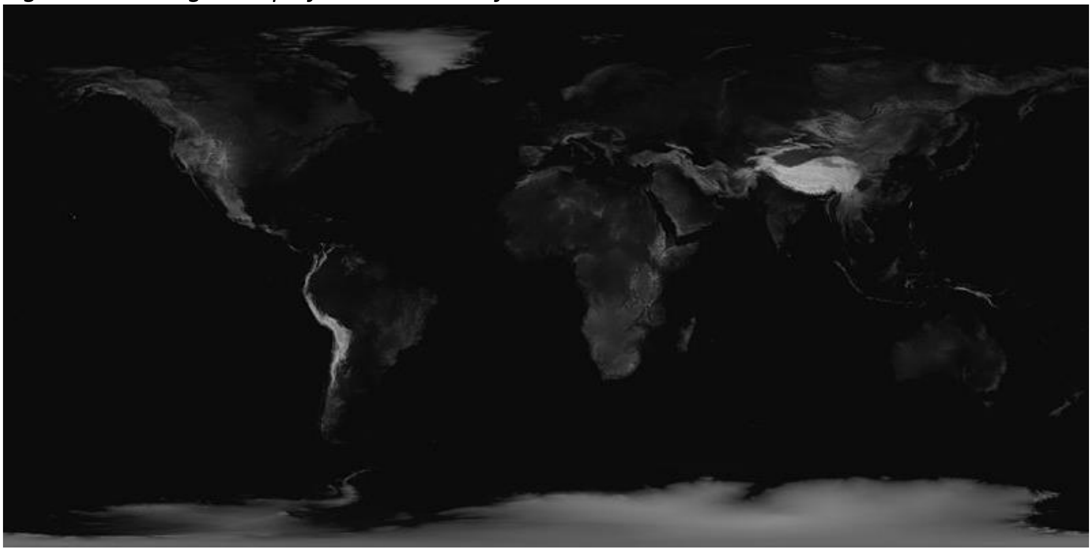


Figure 1.1 shows a height map of the earth's surface, including water and ice ("Topography", 2005). The image is 8-bit greyscale, which represents the height of terrain at a given location by the colour. The whiter a pixel is, the higher the altitude at that location. In figure 1.1 you can see that the more mountainous areas are grey/white, while the sea is represented as black (sea level). Note for example the light area between the Indian subcontinent and China indicating the Himalayas. Despite this height map using sea level for the lowest point (0 above sea level), the relative lowest point can be anything, based on the need of the height map. For example, if the height map is representing a smaller area, then the darkest colour may be the lowest point in that particular area.

Gamito & Musgrave (2001, p. 1) state that the downside to using height maps for terrain representation is that 'overhangs' cannot be represented. This includes caves, the inside of buildings and bridges. This is due to the inability to represent two y values as a single height value on a 2D grid (a height map).

2.3 – Height Map Generation

One way of creating a height map is using procedural generation, which is a method to generate it algorithmically rather than manually. One technique for doing this is to use Perlin noise to generate randomness. Dong, Zhang & Zhang (2010, p.1) describe "Perlin noise" as a method of generating noise with features of randomness. They state that this can be used to generate realistic textures and terrain. Using Perlin noise to generate a procedural height map creates more realistic terrain than simply assigning every pixel to a random value.

There are other similar techniques like Perlin noise that enable the generation of height maps that simulate realistic terrain. However, when it comes to generating a height map that represents *actual* terrain, these techniques can't be used. Instead, it is possible to access archives of real height data such as LIDAR ("LIDAR Composite DSM", 2015). LIDAR is an airborne mapping technique. Archives of this data can be used to create height maps for representation of real places.

When it comes to generating a height map from GPS data, this can be done in two ways. Either the GPS data can be mapped directly to its relative location on the height map using the altitude and location data collected through GPS, or it uses this data to cross-reference with an archive of LIDAR (or other) data. The first technique provides a raw height map created directly from GPS data, whereas the second aims to increase the accuracy of this data before generating the height map.

The latter, while more accurate, has a reliance on third-party data which may be a restriction if the device has no access to the internet.

2.4 – 3D Graphics Representation

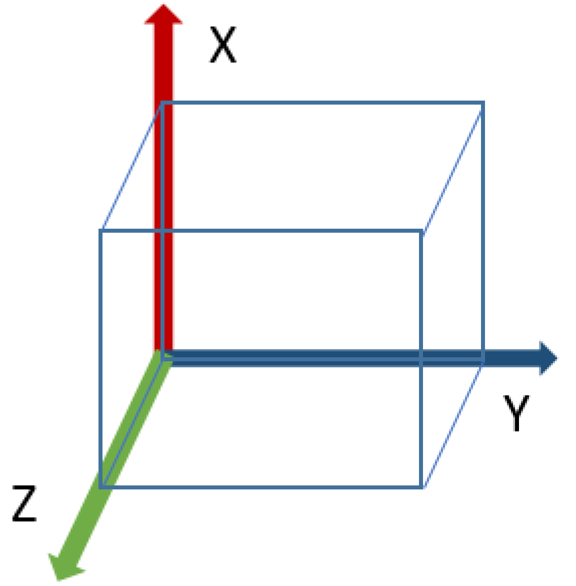
In this project, terrain will have to be displayed to the user as a 3D graphic. This section focuses on the fundamentals of 3D graphics.

There are two main areas of 3D graphics that will be discussed: polygon-based graphics, and voxel - based graphics.

2.4.1 – Coordinate System

The most important part of any 3D graphics is the Coordinate system. Pulli et al (2007, p. 27) states that to define shapes and locations, there needs to be a frame of reference. This frame of reference is the *coordinate system,* or *space.* The coordinate system has an origin, and a set of 3 axes that define directions from this origin. In a three-dimensional space, there are 3 axes; x, y and z. Figure 1.2 below shows these axes.

Figure 1.2



Pulli et al (2007, p. 27) also state that a coordinate system is defined relative to another coordinate system. The one exception to this is the *World Coordinate System*. An example of this is that a table in a room may have its own local coordinate system, with coordinate (0, 0, 0) being, for example, at the bottom, in the middle. The room itself may also have its own coordinate system. Moving the chair in the room does not change the chairs local coordinates, but instead changes the coordinates for where that chair is located in that room.

These ideas are essential to 3D graphics as they are used to define the location of vertices or voxels in 3D space, which will be discussed in the following sections.

2.4.2 – Polygon Graphics

Basic graphics such as cuboids are created by connecting *vertices* to form a shape. A vertex is comprised of coordinates within the coordinate system. For example, a square is 4 vertices that are connected. For a cube, it is 8 vertices connected. Vertices are essentially corners to the shape. This concept is quite simple for basic shapes, however for curved surfaces this can prove challenging. McConnell (2006, p. 136) explains that objects with curved surfaces can be subdivided into polygonal patches. If these patches are small enough, then it is hard to identify that the 3D object is not curved, but instead made up of small, flat polygons.

Polygons are made up in a similar way to a basic cube, with vertices that are connected to create a 'flat' shape. Usually these polygons have 3 vertices, known as triangular meshes (McConnell, 2006, p. 136). By creating lots of polygons, and connecting them to form a shape, you make a polygon mesh.

McConnell (2006, p. 139) goes on to explain that the level of detail in a polygon-based 3D object has many different factors, but ultimately the higher level of detail requires greater computational cost. This can be limited by changing the level of detail for certain objects. For example, small objects or objects in the distance do not need the same level of detail as large or close-up objects. As a rule, the fewer the polygons in the polygon mesh, the lower the level of detail.

2.4.3 – Voxel Graphics

Voxels are another, more recent way to represent graphics in 3D space. Kaufman, Cohen & Yagel (1993, p. 51) state that the voxel is the 3D counterpart of the 2D pixel. This is because it represents a position, just like a pixel, but with 3 axes instead of 2. Instead of connecting voxels together with edges, as done with vertices, voxels are not connected to each other in this way. Instead, voxels are placed next to each other. Gebhart (Gebhart et al., 2009, p. 3) agrees with this, stating that voxels are the 3D analogue to pixels, which are used to uniformly subdivide 2D space into cells. Kaufman (Kaufman et al, 1993, p. 54) further explains that voxels are mapped to either 'black' or 'white', to represent opaque objects or transparent background. Voxel based graphics can largely be compared to 2D Graphics. In 2D graphics, pixels fill up all coordinates in the coordinate system and have associated values that determine colour. Voxels can be thought of in the same way, but in three-dimensional space.

2.4.4 – Comparison of Polygon Vs Voxel

Gebhart (Gebhart et al., 2009, p. 8) explains how voxel graphics require a much larger storage space than polygon modelling, even for an area only 300 metres in diameter. They state that this is because in a voxel grid, every position and value is stored, even if the voxel contains no data. This can be improved by using structures such as Octrees, which gives a compression ratio of 60000 to 1.

Pickton, M. (2012) states that voxels fall short for animations and memory usage without backing from vertexes. They also state that voxels are in general more computationally intensive. This means that while it may not be feasible to use voxels for any large-scale graphics today, this may change as advances are made to Graphical Processing Units.

2.5 – 3D Modelling of Terrain

Terrain can be modelled from a height map using various 3D modelling tools, which use the map to determine the heights of vertices at different points. An example of such a program is ArcGIS. In ArcGIS, raster data (height maps) can be used to generate 3D terrain and visualise this data ("Displaying terrain datasets", n.d.). ArcGIS has many tools that allow the user to visualise, edit, model and query terrain data. The program can also operate in the opposite direction, allowing terrain to be modelled, and then exported to a raster or TIN height map format.

Another method of generating terrain from a height map, more aligned with the goals of this project, is Unity. Polsinelli (2013) explains that Unity is an IDE and a game engine. The Unity website ("The leading global game industry software", n.d.) states that Unity is used on 2.4 billion unique mobile devices, and that 34% of the top 1000 free mobile games are made with Unity. Unity abstracts the complexities of managing 3D assets and makes creating games simpler.

Unity has a feature that allows height maps to be used to generate the terrain within the game. Not only this, it also includes height tools (“Height tools”, n.d.) to make direct edits to the terrain. The height tools operate in a similar manner to painting tools found in image editors, making it simple and easy to create terrain for direct use in games and applications that are built using Unity. As Unity is cross platform, these applications can then be used on iOS, Android, Windows, and even in-browser using Unity Web Player (“Build once, deploy anywhere”, n.d.).

3 – Requirements Elicitation Through Primary Research

3.1 – Background and Objective

Anyone wishing to develop a new product or service would be wise to ensure that the development process includes meaningful input from potential future users. Market Research is one of the tools used to illicit information from a relevant audience. This will direct the development of a concept, highlighting which end benefits people will look for, what product features are important and what features are 'nice to have', that is to say they are appealing but not essential.

The development of this application was guided by a small, quantitative survey among a relevant audience, with the following objective:

Provide meaningful information about potential users' needs and wants from a sports application on their mobile device, and specifically establish what end benefits users prioritize and which features best serve those end benefits.

The results of this research have been used to determine the requirements of the application itself and guide the development process described in this paper.

3.2 – Methodology

The survey was conducted using an online questionnaire of about 10 minutes in duration. Respondents were invited to take part by means of a simple link, which led them to a questionnaire of mainly closed questions, using Google Forms as the survey platform. No financial incentive was offered for participation.

3.2.1 – Sampling

The survey was conducted among a relevant sample of mobile phone users. Due to budgetary constraints, the more common approach to recruit a sample of respondents, representative of all UK households was beyond the scope of this study. Instead, respondents were sampled using a technique called 'snowballing', whereby a small group of initial respondents were all asked to pass on the questionnaire link to people they know who may be eligible to take part.

To draw maximum benefit from users' recent experiences with a broad range of mobile phone applications it was decided not to limit the survey to people who do snow sports, but rather to invite anyone who currently uses mobile applications and who would be interested in using a sports application. As such the survey was conducted using the following eligibility criteria:

- UK residents
- 16 years and over
- Owners of mobile phone
- Using at least some mobile phone apps
- Would be interested in a snow sports application

3.2.2 – Survey

The survey is split into four different parts;

- Basic information about the user
- Sports applications, the importance of different features and benefits
- Snow sports applications, and specifically the concept
- Snow sports related user information

Basic Information

The first section covers basic user information and classification details, such as age, sex, and what brand/type of mobile phone they have. This will provide a context and help to find correlations and extrapolate the information.

Sports Applications

Questions about sports applications in general, to provide a view on what aspects are important to the consumer.

Snow Sports Application Concept

The survey will then explain the application idea, and ask more specific questions, such as:

- How appealing they find this app idea
- How credible it is
- How unique it is
- How likely they would be to use it

Snow Sports Related User Information

The survey will ask for some more information about the user, such as:

- Whether the user does any snow sports
- What their skill level is

The reason for saving the questions about snow sports until the end of the survey is so that the user answers the sports app questions without any knowledge that it is about snow sports, and potentially removing bias.

The survey results helped determine what *end-benefits* users think are important and direct the development of features that can deliver those benefits and contribute to how the user interface should be designed.

3.3 – Findings and Analysis

The survey was completed by 41 people (at the time of writing) aged between 17 and 68. This sample size was sufficient to obtain directional findings, across a wide spread of ages and skill levels. The results of the survey provided some pointers for the application.

Figure 3.1 – What phone do you have?

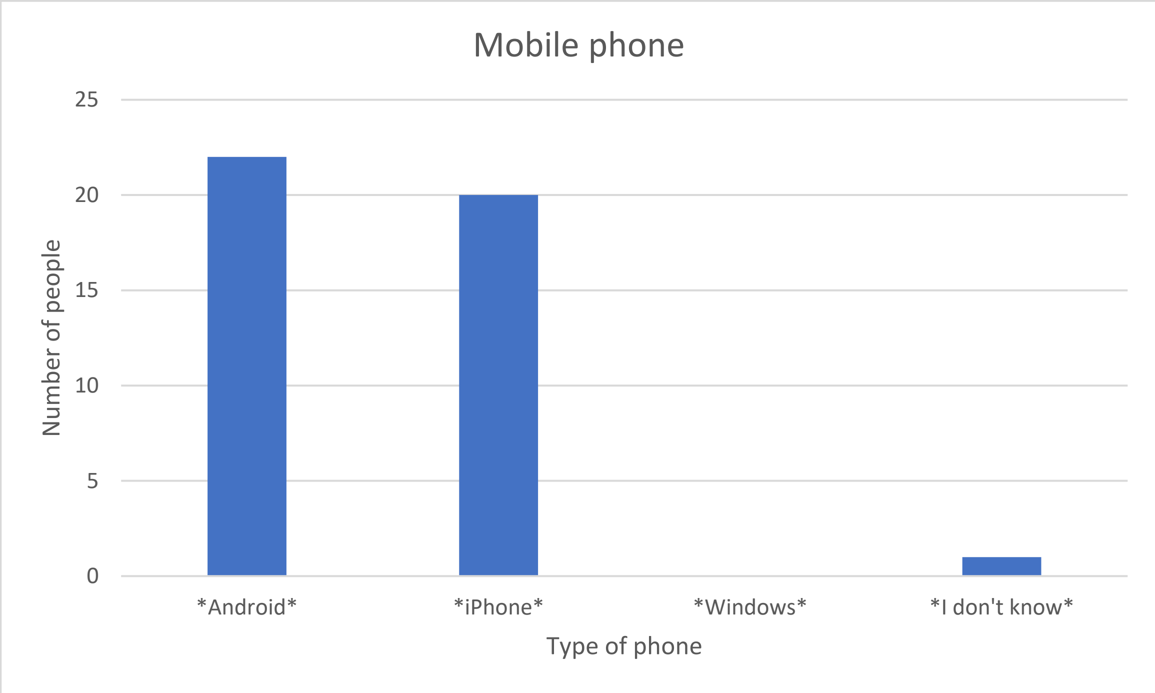
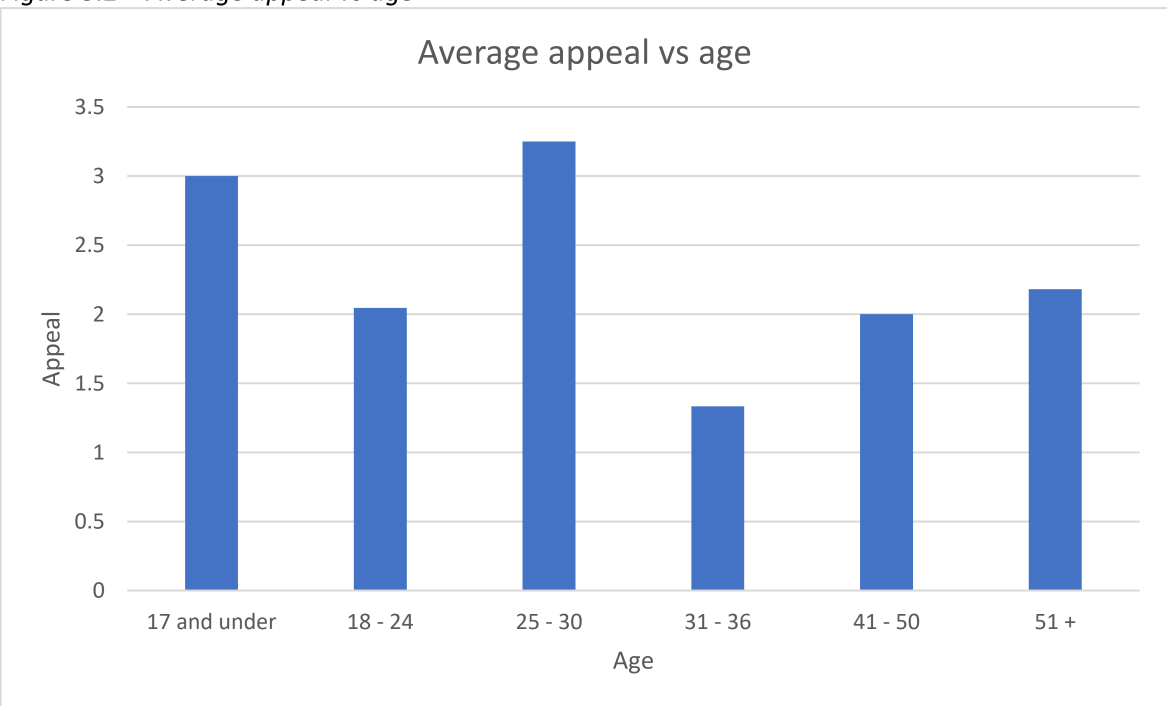


Figure 3.1 shows that the app could be built for either Android or iOS, being nearly tied in use. Android was chosen as the platform for this project simply because a slightly higher percentage of respondents chose it. Alongside this, the application is easier to develop for Android since, unlike iOS, it doesn't require a license to publish to the Play Store. The developer also has previous experience developing for Android, which will help speed up the development process.

Figure 3.2 – Average appeal vs age



Figures 3.2 and 3.3 show that the app has slightly more appeal to younger audiences, and those claiming to be more advanced at winter sports. This could be explained by the younger generation being more tech-savvy and therefore likely to use a mobile application, and those with a higher ski/snowboard level being more likely to want to track their progress. Please note that, due to the small sample size, these findings should be viewed as directional only.

Figure 3.3 – Appeal vs skill level

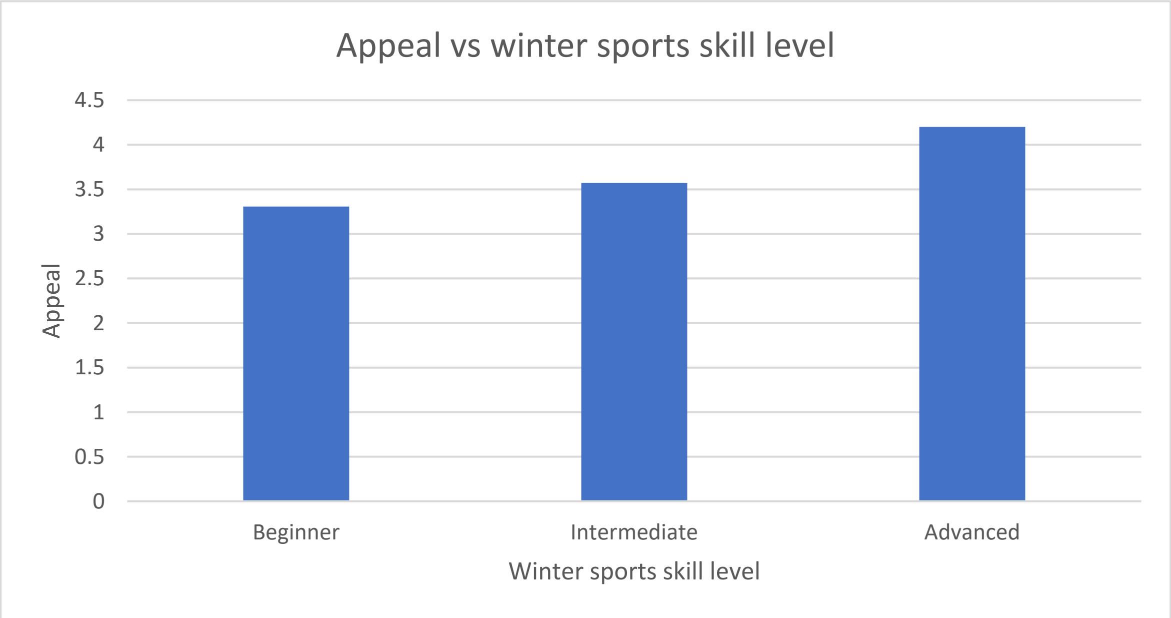


Figure 3.4 – Importance of features in current sports applications

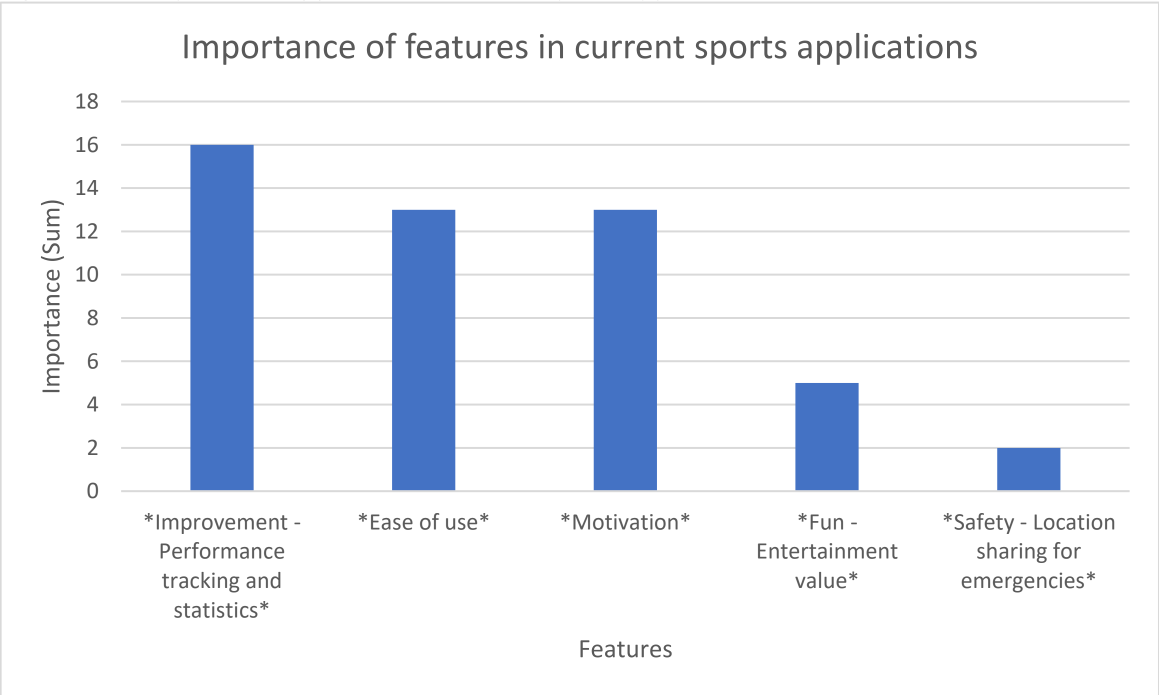


Figure 3.4 shows which features current users of sports apps most important to include. Because the question wasn't specifically about snow sports apps, these results include the opinions of those that haven't done any snow sports. The results show users value performance tracking, ease of use, and motivation. These will therefore be given higher priority in the development of the app, while leaving entertainment, value and safety 'nice to have' extra features, not crucial for the appeal of the concept.

Figure 3.5 – Average benefit importance

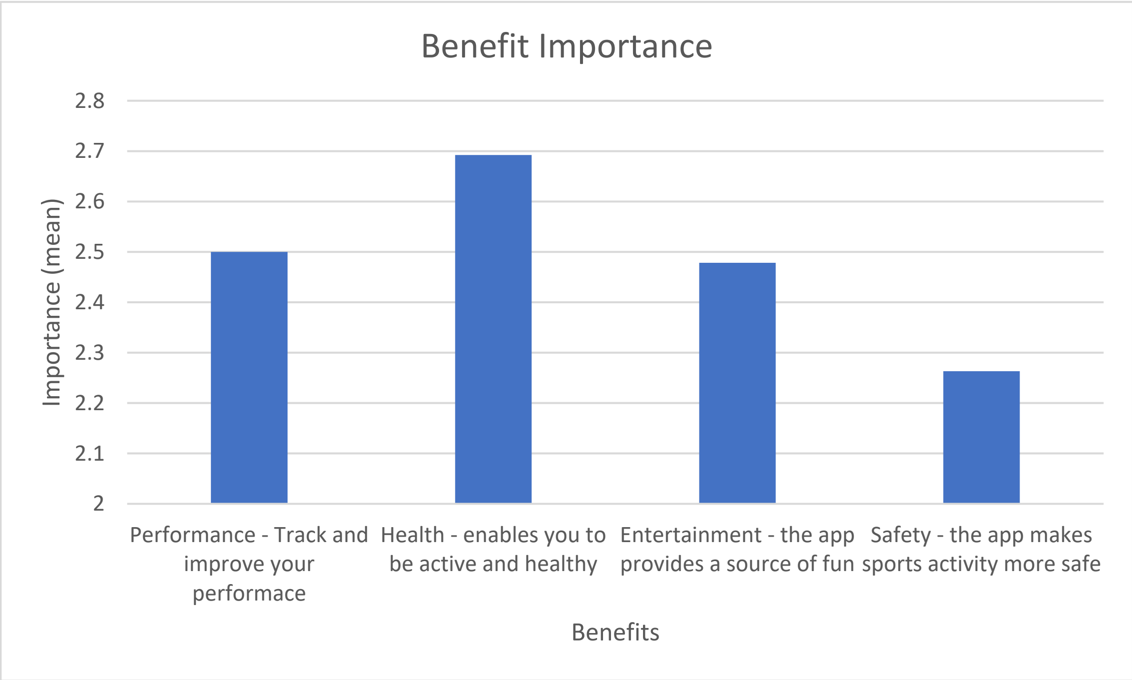
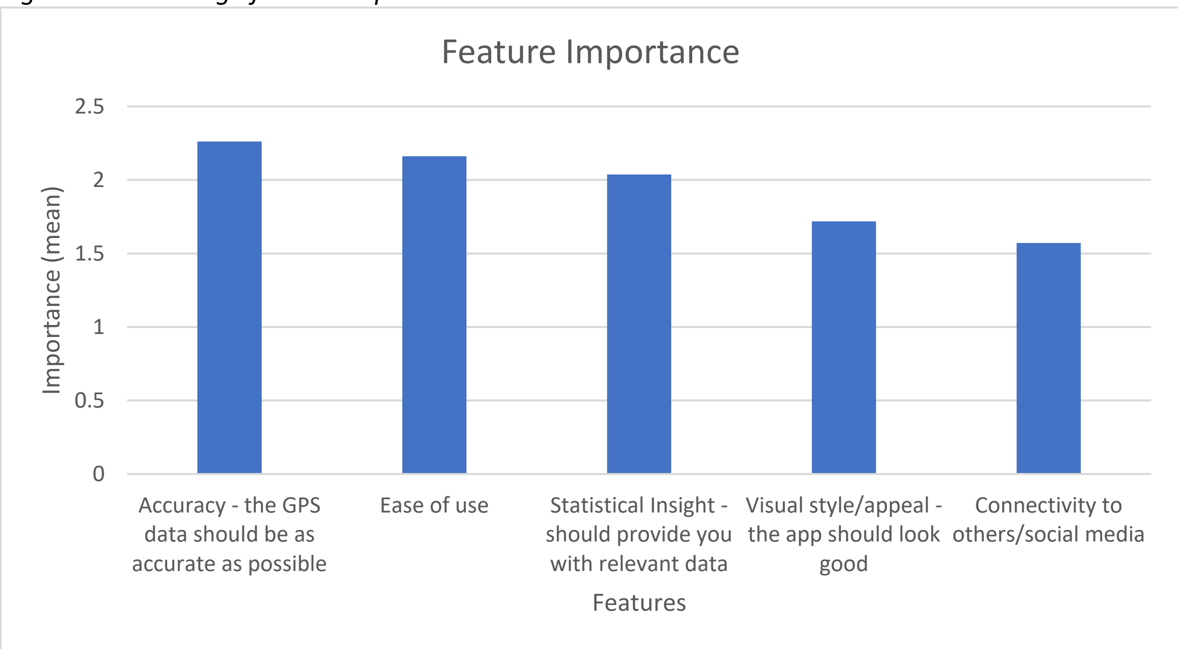


Figure 3.6 – Average feature importance



The last two graphs (figures 3.5 and 3.6), show that users' value accuracy and ease of use to be the most important features to be implemented (although other factors also seem to have appeal), and that the main end benefit for using the app would be health related.

3.4 - Summary of Requirements

Establishing the requirements for the project is a crucial step and determines the process and implementation for the rest of the project. Although this project does not have a client, the survey was conducted to gain an insight into the typical consumers' values in similar projects. The requirements will be split into two parts: functional and non-functional requirements.

The requirements are labelled with an ID so that they can be referenced throughout the project. The requirements are also given a priority, showing the importance of the requirement. In the next section, the requirements will be used to determine the plan for implementation.

3.4.1 - Functional Requirements

ID	Priority	Description and justification
R1	1	The application will track and record the users' position. This is required to collect the data to later be used with the 3D model generation. Without this, the main functionality of the application could not work.
R2	1	The application will generate a Digital Elevation Model from the stored GPS data. This will later be used to generate the 3D model.
R3	1	The application can generate 3D models based on the Digital Elevation Model. This is the core functionality of the project, and all further requirements will depend on this.
R4	2	The user can toggle whether the app is recording their location. This is an important requirement, as the user should have control over whether their location is being tracked. This promotes trust between the application and the user.
R5	3	The user can rotate the 3D model. Without this, the user may not be able to see all aspects of the terrain.
R6	3	The application can continue to record GPS data while the phone is locked. This is important as the users will likely have their phones in their pocket or bag for long periods of time while using the application.
R7	4	The app should provide the user with performance tracking. This will entail the app displaying relevant statistics that allow the user to identify and improve their performance.
R8	5	The app should display the total time spent collecting GPS data. This will include an average function, displaying the time spent per day. The survey concluded that health and motivation were important benefits. By including these statistics, the user can track how much time they spend active and increase the motivation to do more.
R9	2	An accuracy improvement technique is used when generating the 3D model. The survey concluded that accuracy is important to users. The 3D model generation or GPS collection should adopt some technique that improves the accuracy of the 3D model. This is crucial as GPS data collection from handheld devices is notoriously inaccurate.

3.4.2 - Non-Functional Requirements

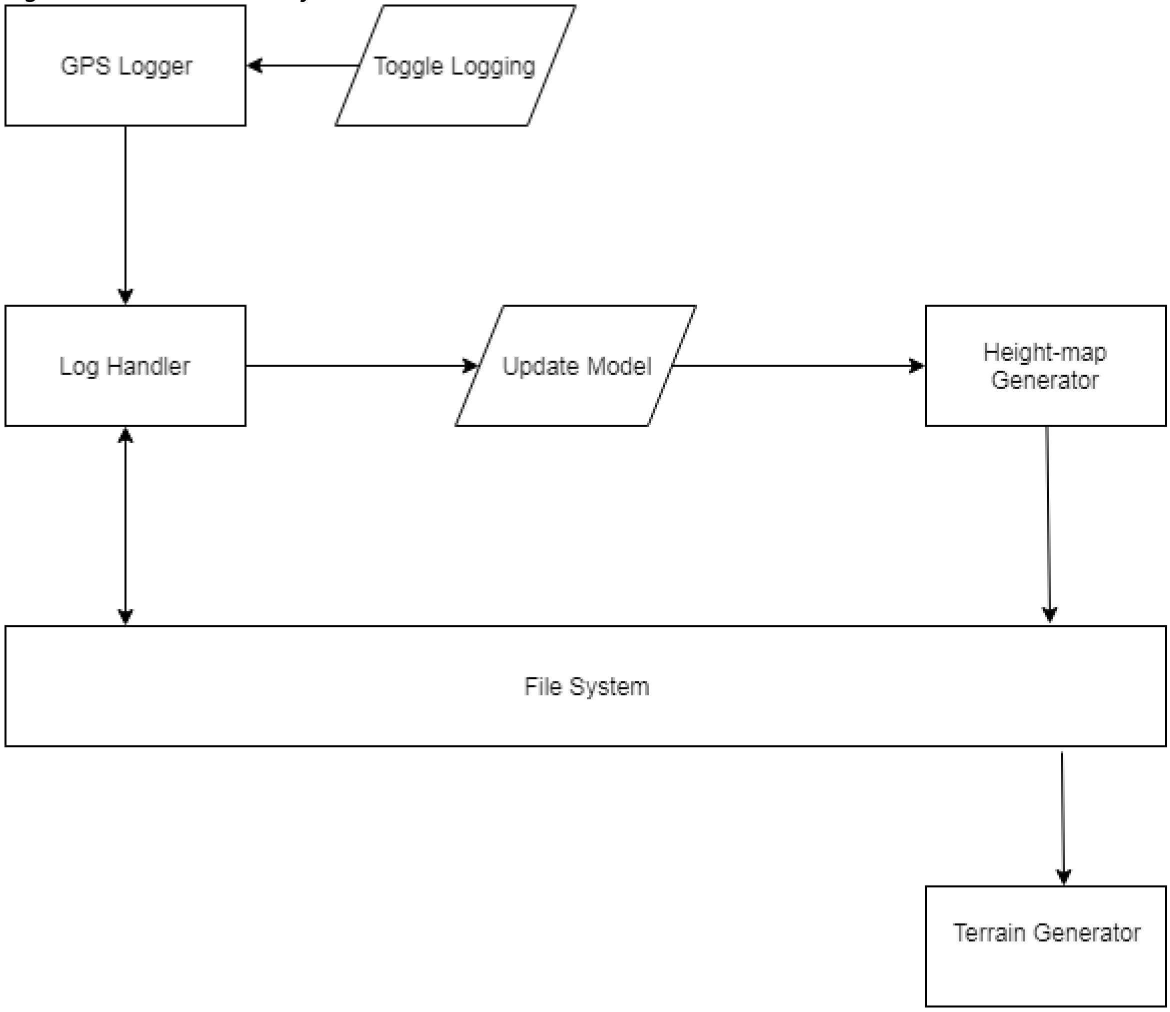
ID	Priority	Description and justification
R10	1	The application should have an intuitive and user-friendly interface. This is important as the application is aimed at consumers rather than professionals. This can be tested using primary market research during the project.
R11	1	The application should be developed for Android phones. The survey concluded that Android and iPhone are the most used mobile phones, and Android was selected as the developer has previous experience developing for this platform.
R12	2	The application should run smoothly on modern hardware. The average Android phone created in the last 3 years should run the application with no performance problems. The application is aimed at general users, and so high-tech hardware should not be required.
R13	1	None of the functionality should require any attachments to work. Extra attachments could be added to improve accuracy/results, but they should not be necessary to achieve the basic functionality.

4 - Design

4.1 - Program Design

Figure 4.1 shows the flow and structure of the designed architecture. It is encapsulated into three parts; the GPS Logger, the Height map generator and the terrain generator.

Figure 4.1 – Architecture flow



4.1.1 - GPS Logger

When the user clicks the 'Toggle Logging' button, the GPS Logger is activated, checking and saving the users GPS location. This makes use of a Log Handler class that stores this data to the persistent file path of the application on the users' device. The Log Handler can both read and write the GPS data to and from the file system.

4.1.2 – Height Map Generator

If the user presses the 'Update Model' button, Log Handler is called again to fetch all the previously stored GPS data. This is then passed into the Height map generator. The height map generator uses this GPS data to create a height map that reflects the users GPS location, and the relative altitude of the user represented by greyscale. This height map is stored in the file system, to be used later by the terrain generator.

4.1.3 – Terrain Generator

The terrain generator will also be called upon clicking ‘Update Model’. However, it will only run after the Height map generator, to ensure that the terrain will include all the latest GPS data collected by the device and GPS Logger. The terrain generator will use the newly generated height map to call Unity’s inbuilt terrain generation functionality that can generate terrain directly from a height map. This will load the terrain directly into the scene and display it to the user.

4.1.4 – User Interface

Figure 4.2

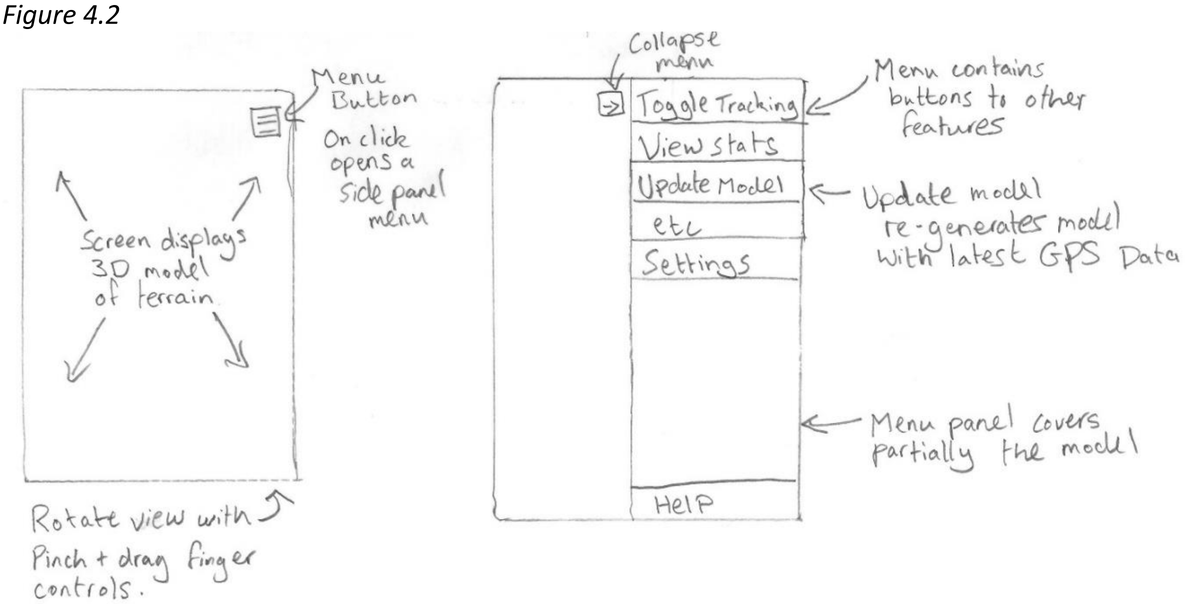
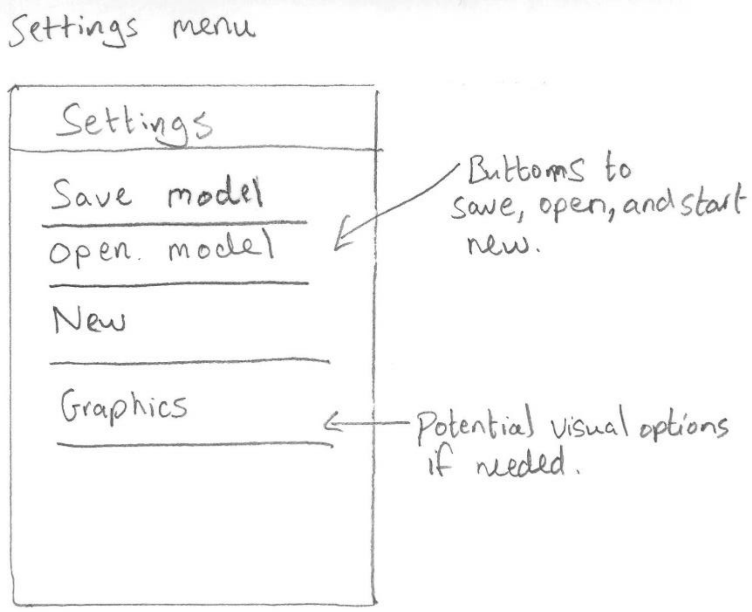


Figure 4.3



The user interface for the program will have the 3D model as the opening page, showing the 3D generated terrain from any previously travelled terrain. This was chosen so that the app can act as a 'map', that when opened immediately displays this information without having to go to another menu.

In the top right of the screen is a hamburger menu that when clicked, creates an overlay menu (covering half of the screen) that displays all other options. These options include:

- Toggle tracking button
- View stats
- Update model (with latest GPS data)
- Settings
- Help

The original hamburger menu button is also replaced with a collapse button, which when pressed, minimises the menu.

Each option that requires a new menu to be displayed will open a full screen menu with all the relevant buttons displayed.

The reason for these user interface design decisions was to create an interface that has the 3D model as the main focus, which is always displayed (even if slightly covered at times). The only time it is totally covered is when the user navigates far into the menus.

The user interface will be built entirely using Unity's inbuilt UI Components. Unity has an easy to implement system for creating simple and effective user interfaces. Each UI component can have a script attached, which can be triggered by events such as button clicks. This allows the UI to directly call the required components.

4.1.5 – Camera

The camera in this context is the 'view' of the generated terrain within the application. By allowing the user to move the view around, they can see more specific details of the generated 3D terrain. This is very important for usability. The camera will be at a set distance from the terrain and controlled by swiping the screen. For example, if the user swipes their finger to the left, the camera will rotate around the terrain to the right, allowing the user to see the terrain from a different perspective. A pinch should also allow the user to zoom in closer to the terrain, and vice versa.

4.2 – Overall Design

The application was designed as described above to use a more modular approach. Each component is a separate part of the system, with as few dependencies between modules as possible. Each module can be tested and changed without causing significant consequences within the other modules. This allows better dependency management, reusable code and improved reliability.

4.2.1 - Android Development and Chosen Technologies

Unity will be used for the development of the application for Android. This also requires the JDK (Java Development Kit) to allow the application to be built and tested on an Android device. Visual studio will be used to edit all scripts, done in the C# programming language. This allows the use of 'solutions' for storing all relevant code for the project and making use of Visual Studios built in Git functionality, giving access to version control directly from the IDE.

4.2.2 - Data Formats

The data format for the GPS data will use NMEA. More specifically, $GPGGA. This allows the latitude, longitude and altitude to be stored, along with some more useful data such as timestamps and accuracy. This was chosen due to this data format encapsulating all necessary data for the terrain generation functionality, along with the fact that NMEA is the global standard for GPS data.

Longitude and latitude are reported in degrees, minutes and seconds; however, altitude is reported in meters. To ensure that the data is parsed and displayed accurately, it will be necessary to convert the longitude and latitude to meters. An issue that comes with this is that locations far from the 0 latitude and 0 longitude points will likely result in an integer overflow error, or at the least a loss in application performance due to the large integer values this would create. To counteract this, the user will be able to calibrate the app at any location, which will make meter values represent distance from this calibration point. This calibration will happen automatically when the user starts the GPS tracking functionality, to make the application easy to use.

4.2.3 – Height Map Generation

To generate the height map from GPS coordinates, it is important to first determine the limits/boundaries of the terrain, as this will be used to determine the size of the image. The highest/rightmost and lowest/leftmost coordinates will be used to determine the width and height of the height map, mapping them in respect to 0 by calculating the difference.

For example, with the following coordinates;

Lowest/leftmost coordinate: (865, 1467) becomes (0, 0)

Highest/rightmost coordinate (1500, 2673) becomes (635, 1206)

This would leave us with a height map of height 1206 pixels and width 635 pixels, assuming each pixel will represent a coordinate in meters at a 1:1 ratio. This ratio could be dynamically changed, making it less accurate as the range gets bigger to save performance cost and time, or manually changed by the user in the settings menu.

The application will then procedurally loop through the GPS coordinates, applying the altitude value to the equivalent raster coordinate on the height map. It may be necessary to apply this altitude value to neighbouring raster coordinates to make the 3D model more visually appealing, as without this the travelled path may look too thin. Again, this option could be included in the settings menu.

4.2.4 – Testing Strategy

The testing strategy chosen for this project is the Black Box testing method. This is used to test software without concern for *how* they are implemented, instead simply testing if the tested functionality is working as expected. It is used to find usability errors and bugs in categories such as Interface Errors, Behaviour, Performance and Database access. This method of testing was chosen to allow the testing to be completed from the user’s point of view, without knowledge of the internal workings of the application itself. This opens the application to many possible testers in the future.

5 - Implementation

5.1 - Overview

The implementation section will explain the technologies used throughout the implementation, and then proceed to explain the development process from start to finish, detailing the technicalities and problems encountered. It will then evaluate the implementation against the design and discuss any shortfalls.

5.2 - Development Environment

A prerequisite to starting this project was ensuring that the appropriate development tools were available and set up. These include, and are not limited to, Unity, Visual Studio, an Android device.

5.2.1 – Unity

Unity was set up as the core of the project. Unity handles the 3D objects of the app, in this case the terrain, as well as the User Interface. Unity also manages all assets and scripts created and used in the project. Upon creating the Unity Project, default folders are created to hold these assets, scripts and builds.

Unity makes use of Java to allow the built applications to run on both iOS and Android, and so it was also important to install the latest Java Development Kit (JDK). This is especially important, as without it the app could not be tested on a mobile device. A few problems were encountered when installing JDK 9.0. Unity was unable to build the application with JDK 9.0, and so JDK 8.0 had to be used instead. Fortunately, this does not limit how the application is developed, and in the future, it can be rebuilt with JDK 9.0 if needed.

5.2.2 – Visual Studio

Visual Studio was chosen for the IDE. This was easy to setup and comes with lots of useful extensions and features for the development of the code. Visual Studios use of 'Solutions' was used to track and manage all code that it encapsulates, which in this case was all C#. Fortunately, there was no need to investigate the Projects configuration and properties, as this is all handled within Unity, with the code being run as individual scripts.

5.2.3 – Version Control

Git was used as the version control system. More specifically, GitHub. This was chosen due to the in-built Git functionality within Visual Studio, allowing the code versions to be managed directly from the IDE. The other contender was Team Foundation Server; however, Git was chosen due to the developers' previous experience with Git. There are no noticeable benefits to using either for a solo project, and so largely it came down to personal preference.

5.2.4 – Android Device

For general testing and debugging of the application, the app was built to run on a OnePlus One running Lollipop 5.1.1. This is useful as the OnePlus One is not a new device, and therefore shows possible performance issues immediately. The use of the Android Device for constant debugging, rather than purely testing within Unity Development Environment, made it convenient to test the GPS functionality without having to rely as heavily on test/fake data. Instead, live GPS data could be obtained straight through the device.

5.3 - Implementation Process

5.3.1 - User Interface

The user interface was designed with accordance to the design, however with one major change. The original design was for users holding their phone in portrait orientation mode. However, upon creating this it was discovered that viewing the terrain when holding the phone in portrait was user un-friendly and clipped off more terrain than is acceptable. To counter this, the same design was used, but in landscape mode. Figure 5.1 and 5.2 shows the user interface in landscape. This seemed to be much more adequate for viewing larger terrains and gives the user a greater feeling of control when navigating around the terrain with the camera.

Figure 5.1 – User Interface with settings panel open

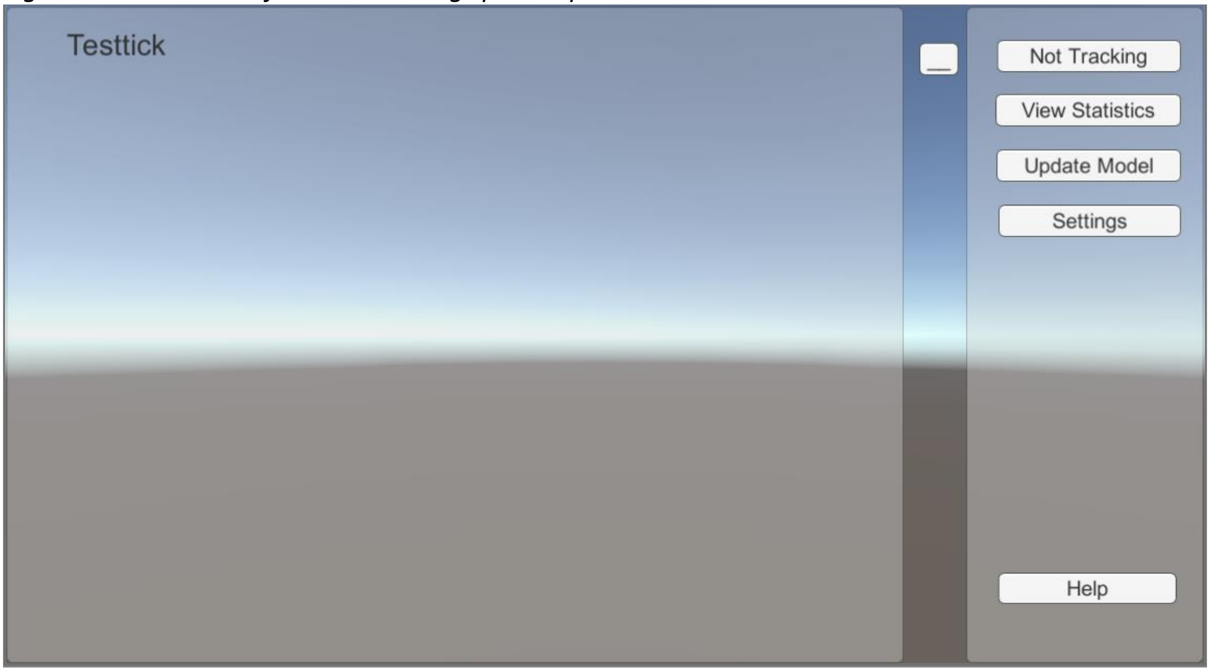
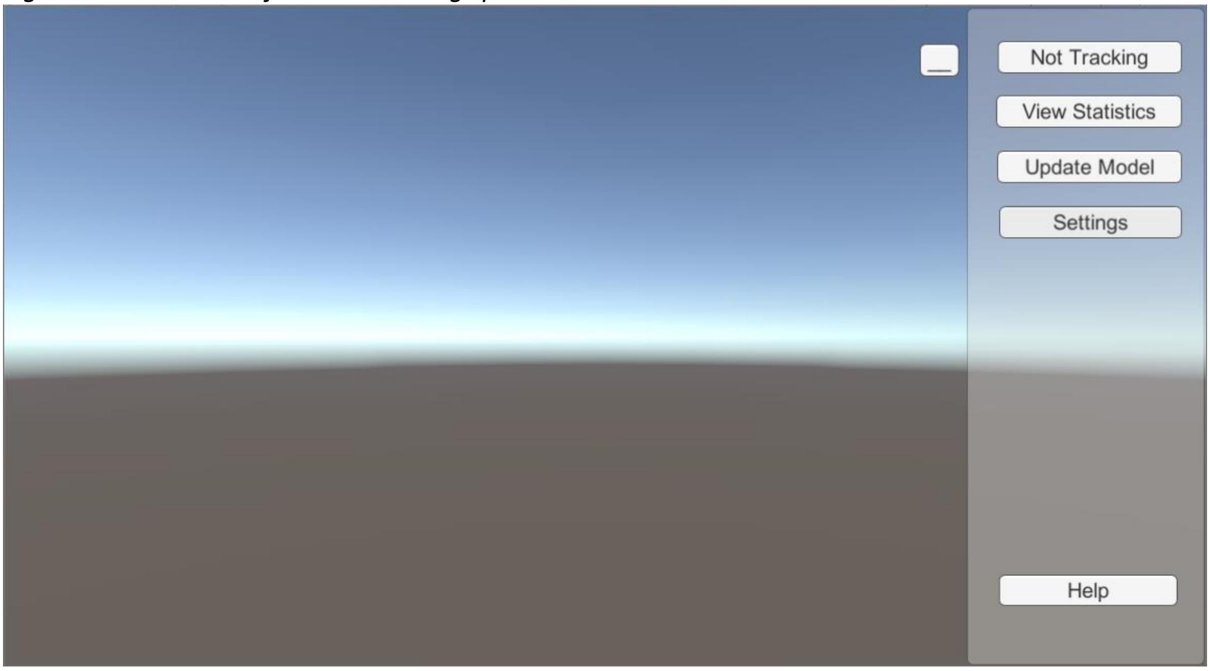


Figure 5.2 – User Interface with settings panel closed



5.3.2 - GPS Logger

Implementation of the GPS Logger was achieved by creating a class ‘GpsLogging’ which handles all GPS logging functionality. ‘GpsLogging’ is attached to a game object in Unity called ‘Scripts’. This is loaded in with the scene and allows the GPS logging functionality to be accessible upon loading the app. Loading the app automatically starts a co-routine that checks the ‘LocationServiceStatus’ and, if it is running, saves the current GPS Location data to a text file stored in persistent storage on the device. By clicking the ‘Not Tracking’ button, the location services are started, and if there are no problems then ‘LocationServiceStatus’ starts running. Clicking the button again then turns location services off, and therefore stops logging the GPS location data to a file.

Unfortunately, there were problems implementing this as a service, and for the GPS Logging to be active, the app must be open on the phone. The device can still be locked, despite the app needing to be open. This should make it acceptable for use while skiing/snowboarding, although future work could be done to make this functionality a service, and therefore enable GPS Logging even while the app is closed or minimised.

5.3.3 - Height Map/Terrain Generator

Generating a height map and terrain was implemented slightly differently to the design. The original intention was to generate a height map, which would then be used to generate the terrain. This was designed as a dependency such that the user would click ‘Update Model’ which would create a height map from the stored GPS Data, and then this would be used to generate the terrain in Unity. However, it was discovered that using a height map for terrain generation in Unity, although usually fine, does not work at runtime. Instead, the terrain can be manipulated at runtime directly without the use of a height map, from the GPS Data. Nevertheless, the functionality that generates a height map was implemented, as this may still be useful to users that want to export this data from the application and use it in any 3rd party application. However, within the app itself, this height map is never used. Instead, the same algorithm that creates the height map also manipulates the terrain directly, in the same fashion. Ultimately this is a net positive, as it removes the dependency on the height map, whilst keeping the benefits of having a height map generator.

The implementation of the terrain generator uses a list of the GPS coordinates, gathered from the LogHandler, to determine the highest and lowest values for the latitude, longitude, and altitude. These values are used to set the relative location for a 2D array. For example, if the minimum latitude is 52.586, then index 0 is 52.586. All other index’s values are relative to this. To avoid using doubles, the latitudes and longitudes are multiplied by an ‘accuracy’ parameter, that can be changed. This converts them to integers and allows one to one relative location in the 2D array.

The program then creates a 1D array and a 2D array. The 2D array is used for creation of the height map, whereas the 1D array is used for the terrain generation. The 1D array must represent a flattened 2D array, and therefore the indexes need to be found by finding the ‘flattened’ index of the 2D array, using the following formula;

Figure 5.3

```
index = (relativelatitude * (xheight + 1)) + relativelongitude;
```

The relative latitude and relative longitude are the 2D array indexes found as described earlier. The xheight is the total distance between the lowest and highest longitudes. This formula finds the appropriate index for the flattened 2D array (1D array).

The program loops through the GPS data from the LogHandler, and stores the relative altitude values, stored as a grey RGB value in the correct array indexes, found as described above. The highest point is stored as white, and the lowest as black. Once this is complete, the 2D array is stored as an image to create a height map, and the 1D array is used with Unity's terrain functionality to set the terrain heights. The terrain is then generated using Unity's CreateTerrainGameObject function.

This implementation differs from the design as the height map created is not directly used to manipulate the terrain. Instead, the terrain is manipulated in a different fashion, that does not require this height map.

5.3.4 - GPS Averaging

GPS averaging was found to be required during the implementation process, as without some form of estimation the generated terrain would look strange. GPS Averaging is the idea that the GPS data points collected by the device were too spaced apart and inaccurate, and so some form of estimation between data points would be required to make the generated terrain sufficient.

Unfortunately, due to time issues and other problems with implementation, this functionality was not completed. The functionality worked, to a certain degree, able to add fake terrain points by averaging the distances and altitudes between two points. This worked by using Linear functions to estimate many locations between two points, and then finding the estimated altitude by using a percentage of the distance between the points, considering the altitude of the points themselves. It did, however, have many bugs. Due to limited time, these bugs were not tested and fixed, and so the functionality was left out. Future work would include finishing this functionality and including it in a future release of the application.

Although unfinished, this functionality was implemented by using linear functions to determine the coordinates between two points. For example, if there is no GPS data between two data points, set 50 metres apart, the application would try to estimate the data points between these two points. The following linear function was used:

Formula 5.4

y = ax + b

Formula 5.5

a = (y2 - y1) / (x2 - x1)

Formula 5.6

b = (y1 - (a * x1))

a can be calculated with formula 5.5, where y1 and x1 are the first set of coordinates, and y2 and x2 are the second set. Once a is calculated, b can be calculated with formula 5.6. With a and b values found, the original formula 5.4 can be used to determine the value of y at any value of x. By looping through the x values between the two GPS coordinates, the program then determines the y values, and uses these values as estimated locations between two GPS data points.

5.5 - Evaluation of Implementation Against Design

In the end, the implementation strayed significantly from the design. The GPS data format used in the implementation was changed from NMEA to basic comma separated values. This allowed the application to store specifically what was needed for the application, with no extra unnecessary information. This does however make the application harder to make compatible with other GPS related software, if decided to do so in the future.

Another area that differed from design was the terrain generation. Originally the idea was to use the generated height map along with Unity's built in terrain generation. Due to this functionality being unavailable at runtime, instead specific coordinates had to be manipulated at run time, making the height map unnecessary.

These changes, considering that not all the functionality was completed, make for quite significant changes from the design. However, the functionality of the application is not affected besides the fact it is behind schedule, and so ultimately these changes may have been positive by allowing a better program design.

6 – Testing

6.1 – Black Box Testing

Black box testing is used to test features without concern for how they are implemented. To do this, a list of tests with expected outcomes, based on the requirements, was created. These tests are designed to test the software's reliability, performance and stability. Some of the tests include expected time that any action should take, to test performance. These tests were performed on a OnePlus One Android device.

Test	**Expected Outcome**	**Result**
Dragging the screen to rotate camera	Camera rotates left/right according to the direction of swipe on touch screen	Fail
Clicking update model to generate 3D terrain	Terrain is generated.	Pass
Clicking update model when there is already terrain	Terrain is updated to reflect any changes	Pass
Clicking toggle tracking button multiple times	Alternates between tracking and not tracking, button text correctly represents action	Pass
Clicking settings button	Alternates between opening and closing settings menu	Pass
Clicking update model creates height map	Height map based on GPS data is generated and stored in persistent storage on device	Pass
GPS Logger stores GPS Locations	While GPS tracking is active, GPS Locations are stored in persistent device storage in a text file	Pass
View Statistics button	Opens a statistics menu and displays relevant information	Fail
3D terrain utilises location averaging	The 3D terrain height map includes 'in-between' data points, determined by averaging locations and altitudes between each GPS Location	Fail
Delete stored height map, then click update model	Height map is regenerated, followed by terrain	Pass
Click update model, time how long terrain generation takes	Less than 5 seconds to generate terrain	Pass – 4 Seconds

6.2 - Evaluation of Results

These tests show that for the most part, the main features have been successful and passed. The application can generate terrain, a height map and log GPS. However, there are still a few features that failed, largely due to these features not being implemented at all. An example of this is the camera function, which has not been implemented. Unfortunately, this means that the application is not usable as a mobile application yet, requiring a development environment to see the resulting 3D terrain with any detail. Another example is terrain location averaging which, as of the time of writing, has too many bugs and is not included in the functionality.

Despite this, the results of the tests show that in most cases the application is stable and reliable, with minimal bugs. The main cause for failed tests is lack of implementation. Future work to finish the remaining functionality should allow these tests to pass.

7 – Evaluation

In this section, the application is evaluated against both the Functional and Non-Functional requirements, specified in section 3.

7.1 – Functional Requirements Evaluation

Due to the objective nature of functional requirements, to evaluate the implementation simply involves checking that the functionality in the requirements is met in the implementation. The following table shows each functional requirement and states if the requirement was met.

ID	Status	Description and justification
R1	Met	The application will track and record the users' position. This is required to collect the data to later be used with the 3D model generation. Without this, the main functionality of the application could not work.
R2	Met	The application will generate a Digital Elevation Model from the stored GPS data. This will later be used to generate the 3D model.
R3	Met	The application can generate 3D models based on the Digital Elevation Model. This is the core functionality of the project, and all further requirements will depend on this.
R4	Met	The user can toggle whether the app is recording their location. This is an important requirement, as the user should have control over whether their location is being tracked. This promotes trust between the application and the user.
R5	Not met	The user can rotate the 3D model. Without this, the user may not be able to see all aspects of the terrain.
R6	Met	The application can continue to record GPS data while the phone is locked. This is important as the users will likely have their phones in their pocket or bag for long periods of time while using the application.
R7	Not met	The app should provide the user with performance tracking. This will entail the app displaying relevant statistics that allow the user to identify and improve their performance.
R8	Not met	The app should display the total time spent collecting GPS data. This will include an average function, displaying the time spent per day. The survey concluded that health and motivation were important benefits. By including these statistics, the user can track how much time they spend active and increase the motivation to do more.
R9	Not met	An accuracy improvement technique is used when generating the 3D model. The survey concluded that accuracy is important to users. The 3D model generation or GPS collection should adopt some technique that improves the accuracy of the 3D model. This is crucial as GPS data collection from handheld devices is notoriously inaccurate.

As shown in the table above, not all the requirements were met. While the core aspects of the functionality were completed, the extra features such as performance tracking (R7) and accuracy improvement (R9) were unable to be completed in time.

7.2 – Non-Functional Requirements Evaluation

Non-functional requirements, being more subjective, are harder to test for. In the table below is a list of the non-functional requirements and their status. Originally, a short survey was going to be conducted that allows the user to test the app and give their feedback on the User Interface and performance of the application. However, due to the number of bugs and unfinished functionality, this data would be redundant in the applications current state. The user interface design itself may be good, but as many buttons and features are either not implemented or implemented in a very 'beta stage' way, the feedback would likely consist of confusion about how the application works.

ID	Status	Description and justification
R10	Not met	The application should have an intuitive and user-friendly interface. This is important as the application is aimed at consumers rather than professionals. This can be tested using primary market research during the project.
R11	Met	The application should be developed for Android phones. The survey concluded that Android and iPhone are the most used mobile phones, and Android was selected as the developer has previous experience developing for this platform.
R12	Met	The application should run smoothly on modern hardware. The average Android phone created in the last 3 years should run the application with no performance problems. The application is aimed at general users, and so high-tech hardware should not be required.
R13	Met	None of the functionality should require any attachments to work. Extra attachments could be added to improve accuracy/results, but they should not be necessary to achieve the basic functionality.

Due to the reasons stated above, R10 is not met. In its current state, users are not expected to find the user interface intuitive and user friendly. As for performance (R12), the requirement is met. Generating terrain, and all other areas of the interface, are quick and responsive when running on a OnePlus One android device. Any newer hardware is likely to have even better performance.

7.3 – Evaluation Against Requirements

Evaluation against the requirements stated at the start of the project shows that although not all functionality was finished, the core functionality is complete (GPS tracking, height map generation and terrain generation). There are however a few features incomplete, such as GPS averaging and camera controls. It is fair to say that although unfinished as an app, future work could finish the remaining functionality, and due to modularization, the current functionality could be exported and used in any other software application.

7.5 – Evaluation Against Existing Applications

Existing systems that log the users GPS location to display this information in a visual way have been around for a while. These applications are largely used for navigation, but some are also used to track historical location for review later (health and sporting applications). The main difference between this application and other GPS applications is in the way the data is displayed. Traditionally, the GPS path is displayed on a pre-existing two-dimensional map. This is different in this application as the map itself is generated purely from the GPS data, and displaying this information in three dimensions. This is a significant change that proves that the application successfully contains a degree of ingenuity and provides new features.

7.6 – Future Applications and Changes

Although unfinished, this application could have various future applications, changes and enhancements. Firstly, due to the open source nature of the project, any developers in the future are free to use any functionality within their own applications. The most important feature for this is probably the height map and terrain generators, as they are the most unique part of the application. These functionalities could be used within other software for automatic terrain generation. Potential applications for this app outside of snow sports but within the sports arena could be found in cycling, jogging, hiking, skating etc. Potential applications outside of the sports arena could be found in navigation walks, hunting, training simulations, outdoor activities etc.

As for future changes to the application itself, the most important aspect would be to complete all unfinished functionality and remove any large bugs. Upon completion of this, the application could then have more features developed for it, such as;

- Camera integration for location-based pictures
- Features for different sports
- Emergency mode, automatic distress signals
- Ability to add friends and see their locations

8 - Conclusion

8.1 - Evaluation Against Project Aims

This section evaluates the project against the aims.

The first aim was to c*onduct background research into GPS and terrain generation.* This aim was met by producing a literature review in section 2 that evaluates relevant information and technologies from literature and was used to gather information that could help with design and implementation of the application.

The second aim was to c*onduct a consumer research survey (primary research) to determine the requirements as dictated by potential end users.* This aim was met through the creation of a survey in section 3 (requirements). This survey was completed by 41 people, and the information gathered was used to determine the requirements for the application by analysing the results. Further future research could be conducted to test the functionality of the app through user testing.

The third aim was to p*roduce a software artefact that can track GPS data, and generate a 3D model based on this.* Although some functionality from the requirements were not met, as discussed in section 7 (evaluation), the core functionality as described in this aim was met. The application can track GPS location and generate a 3D model at runtime from this data.

The final aim was to d*iscuss further applications and potential future changes to the software*. The code itself is open source and free to be used by any developer looking to use this terrain generation method in their projects. Further changes to the software was discussed throughout the evaluation and implementation sections. Due to these reasons, the aim was met.

8.2 – Project Management

The design methodology used in this project is the Waterfall model. As this project was an individual project, the waterfall model was adequate without any major changes, and without more complex models being utilised.

The lack of completion of the implementation can be attributed to a lack of in-depth and relevant research in the literature review. This literature review was conducted at the start of the project and used to help conduct the design for the application. However, during the implementation many of the design decisions had to be rethought due to a lack of *thorough and relevant* research. With a better design from the start, based on more relevant research, time would have been saved. This, in turn, led to a loss of crucial time that could have been spent on the implementation process. Any future work conducted on this project will require some more thinking about the continued design and will likely require some refactoring of code to implement any changes.

8.3 – Final Conclusion

This project conducted research and evaluation into terrain generation methods and GPS. A software artefact capable of tracking GPS locations and generating a 3D model of this data was developed. This software was developed following the Waterfall model, with a survey conducted to determine the requirements. The implementation was evaluated against these requirements, meeting many of them but unfortunately not completing everything that was planned. The project was largely successful, with the main downfall being a lack of completion. Future changes were discussed, showing thought into future potential uses for the application, as well as potential uses for the functionality within other applications.

On a personal level, the development of this project has increased the authors knowledge on terrain generation, C# programming, Android development, while also giving valuable experience with time management, market research design, execution and analysis and project planning.

References

Agile vs. Waterfall: A Side-by-Side Comparison. (n.d.). Retrieved from https://www.upwork.com/hiring/development/agile-vs-waterfall/

Build once, deploy anywhere. (n.d.). Retrieved March 15, 2018 from https://docs.unity3d.com/Manual/terrain-Height.html

Displaying terrain datasets in ArcGIS. (n.d.). Retrieved from http://desktop.arcgis.com/en/arcmap/10.3/manage-data/terrains/about-displaying-terrain-datasets-in-arcgis.htm

Dong, W., Zhang, X., Zhang, C. (2010). Generation of Cloud Image Based on Perlin Noise. 2010 International Conference on Multimedia, 1, http://dx.doi.org/10.1109/MEDIACOM.2010.77

Gakstatter, E. (2015). *What Exactly Is GPS NMEA Data?* Retrieved from http://gpsworld.com/what-exactly-is-gps-nmea-data/

Gamito, M. N., Musgrave, F. K. (2003). Procedural Landscapes with Overhangs. Retrieved March 15, 2018 from: https://www.researchgate.net/publication/2948853_Procedural_Landscapes_with_Overhangs

Gebhardt, S., Payzer, E., Salemann, L., Fettinger, A., Rotenberg, E., & Seher, C. (2009). Polygons, point-clouds and voxels: A comparison of high-fidelity terrain representations. In *Simulation Interoperability Workshop and Special Workshop on Reuse of Environmental Data for Simulation—Processes, Standards, and Lessons Learned.*

GPS. (n.d.). Retrieved from https://developer.android.com/things/sdk/drivers/gps.html

Height Tools. (n.d.) Retrieved March 15, 2018 from https://docs.unity3d.com/Manual/terrain-Height.html

HISTORY OF NMEA (n.d.). Retrieved from https://www.nmea.org/content/join_the_nmea/history.asp

Kaufman, A., Cohen, D. & Yagel, R. (1993). Computer. *Volume Graphics, 26*(7), 51-64. http://dx.doi.org/10.1109/MC.1993.274942

LIDAR Composite DSM – 1m. (2015). Retrieved from https://data.gov.uk/dataset/lidar-composite-dsm-1m1

McConnell, J. J. (2006) *Computer Graphics, Theory into Practice*. Massachusetts, Jones and Bertlett.

Pickton, M. (2012). *What is a voxel, anyway? Voxels vs. Vertexes in Games.* Retrieved from https://www.gamersnexus.net/gg/762-voxels-vs-vertexes-in-games

Polsinelli, P. (2013) *Why is Unity so popular for videogame development?* Retrieved from https://designagame.eu/2013/12/unity-popular-videogame-development/

Powell-Morse, A. (2016). *Waterfall Model: What Is It and When Should You Use It?* Retrieved from *https://airbrake.io/blog/sdlc/waterfall-model*

Pulli, K., Aarnio, T., Miettinen, V., Roimela, K., & Vaarala, J. (2007). *Mobile 3D graphics: with OpenGL ES and M3G.* Elsevier.

The leading global game industry software. (n.d.). Retrieved March 15, 2018 from https://unity3d.com/public-relations

Topography. (2005). Retrieved March 15, 2018 from the Nasa website: https://visibleearth.nasa.gov/view.php?id=73934

What is Agile model – advantages, disadvantages and when to use it? (n.d.) Retrieved from http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/

3D terrain using a heightmap. (2014). Retrieved from http://www.bluemoonproductions.nl/tutorials/3d-terrain-using-a-heightmap/

Appendix

Project Initiation Document

School of Computing final year project

Jaco van Cranenburgh

PJE40

Project Initiation Document

Exploring the potential of Digital Elevation Model generation from GPS in consumer products.

Project Initiation Document

1. Basic details

Student name:	Jaco van Cranenburgh
Draft project title:	Using GPS to create 3D Models of terrain in a sports mobile application.
Course:	Computer Science
Client organisation:	N/A
Client contact name:	N/A
Project supervisor:	Jonathan Crellin

2. Outline of the project environment and problem to be solved

My project is to make a sports app, making use of inbuilt GPS in mobile phones to create 3D models of the terrain. I will research Digital Elevation Model generation techniques, and create an application that will track the users GPS data, and create a 3D model of the terrain with altitude considered. This will then be used as the base to explore ideas for extension.

My 'client' (target audience) will be skiers and snowboarders. This is because they operate in mountainous areas, and therefore will have the most use for a 3D model of terrain. The purpose is to create a prototype that has features not currently found in consumer sports apps.

Base implementation (Requirements)

- Create an intuitive user interface
- Use the phones GPS to track and store location history
- Create digital elevation models from the recorded GPS data
- Generate a 3D model from the height map data
- Create more accurate models by using various DEM interpolation techniques

The idea is that, with enough time spent skiing, the user will have a complete and semi-accurate 3D model of the ski resort/mountain. I will develop this with extendibility in mind, ensuring that it can be built upon with ease.

Potential extensions to be researched

- Adding GPS metadata to a camera feed via the app
- Record specific ski runs, compare duration with previous runs
- Communicate GPS between devices to display other users locations

3. Project deliverables

My project will contain 6 main artifacts.

- Initial Research Document into android application development. This will include research into heightmaps, and any relevant APIs or systems already in place for use in mobile applications in relation to 3D Modelling and Digital Elevation Models. I will use this to decide which tools and platforms to use when creating the prototype. I will also research appropriate design patterns and methodologies to aid in development.

- Design specification for the user interface. This needs to be user friendly, as the target audience of the app is for consumers rather than professionals.

- A Literature review into Digital Elevation Models, GPS, and encompassing techniques for generation when using GPS data. This will include the various types of DEM's, GPS collection techniques, and formats for these data's. I will be looking in detail at the various ways to increase a DEM's accuracy, as I suspect that will be the biggest hurdle in implementation.

- The GPS tracker. This will be taken skiing by volunteers to gather test data to aid in the creation of the app. This will then later be implemented into the application. If this is not completed in time for the ski period, then I will have to use online sources for test data. This would likely not be as appropriate for my project.

- A functional mobile application that both tracks GPS data, and converts it to DEM, and then display it as a 3D model. This will be the final product, and should be intuitive and easy to use.
- Test strategy document. This will detail how I will be gathering test data, and how it will be used. This will include an ethical examination checklist and consent form.

- A project report. This will summarize and highlight my project process.

4. Project constraints

There are numerous possible constraints with my project. Time will be a limiting factor. I will have to create a prototype GPS logger before December, to give adequate time to find a volunteer to collect test data. This will give me from January onwards to use this data to test my app. After this, I will be working toward the project deadline, April 27th.

Accuracy will be a big constraint. Due to the nature of GPS in mobile phones, my solution will inherently be somewhat inaccurate. My solution will likely rely on the user going over the same areas multiple times. The more times, the more accurate. I will research into methods for improving this accuracy, or at least improving the *relative* accuracy when compared to other sections of the same mountain.

5. Project approach

My background research will largely be on the technologies involved with my base implementation. This includes height-map data, GPS, relevant 3D modelling software such as ArcGIS, and Android development. I will need to determine the most suitable methods and technologies for my application, such as Language, IDE and engine/program for integrated 3D model generation.

Due to the nature of my project being explorative, I will not be able to plan ahead completely, and the focus of my project may change. I may hit unexpected dead ends in my research, or decide certain functionalities I am trying to implement are not needed. Based on this, I will use the Agile Development methodology. This will allow me to reprioritize or update my plan at any point in my project.

6. Facilities and resources

I will likely need to make use of various currently unknown devices to aid me in my research and experimentation. This will include multiple android devices for testing purposes, and possibly a handheld GPS device for testing with more accurate GPS data. These will largely be acquired from the School of Computing Equipment Warehouse (SKEW).

For software, I will likely have to use ArcGIS during my experimentation. I am able to access this software through the University of Portsmouth, and can either use it on a shared University PC, or access the software from my personal PC by making use of their software share, AppsAnywhere.

I will also have to communicate with the Snowportsmouth Society in order to find people willing to volunteer in collecting GPS data for testing purposes.

7. Log of risks

Risk	Impact	Probability	Mitigation	Indicator
Unable to find volunteer to collect GPS data.	Limited test data, fake instead of real data.	Medium	Use of generated test data or GPS data from an online source. Less accurate, but should be enough to test my app.	If I haven't found a volunteer when nearing the December holidays.
Prototype not created in time.	Possible inability to collect test data. Lack of GPS collecting function in app.	Low	Same as above	No prototype/lack of GOS collecting functionality when nearing December holidays.
Time constraints				
Unable to obtain android devices from SKEW	Inability to test the app on different devices.	Low	Ensure I book these devices early from the SKEW. Find other means such as borrowing a friend's device.	SKEW booking is full.
Lack of relevant research material	Inability to research the topic thoroughly.	Low	Do more experimentation, and rely less on research.	
Height map/Model generation producing vastly unexpected results from real data.	Inaccurate functionality, time constraints.	Medium/High	Adjust project plan to allow more time spent on height map generation, bug fixing, tweaking.	Height map/model generation produces unexpected results.

8. Starting point for research.

The research will include looking at:

- ArcGIS documentation
- Unity documentation
- GPS systems/documents
- Height map/Digital Elevation Models
- Android development
- Current GPS oriented sports apps such as Strava

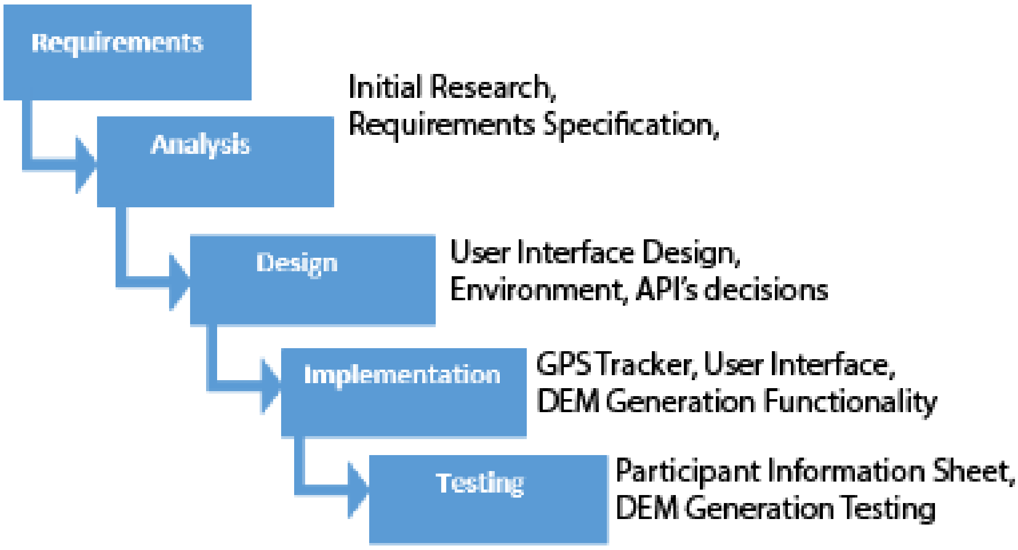
I will find resources for these topics through the university library Discovery tool and Google Scholar. Alongside this, I will use my initiative to find examples of related or similar examples to study, such as other mobile applications.

A few starting examples:

- *Kaplan, E. D., & Hegarty, C. (2006). Understanding GPS : principles and applications. Boston, Mass. ; London : Artech House.*
- *Transactions of the ASAE. VOL. 41(4): 909-916 . (doi: 10.13031/2013.17247) @1998 R. L. Clark, R. Lee*

9. Breakdown of tasks

These are my main tasks, alongside the Waterfall model.



Due to my constraints, such as the time constraint of the GPS Logger for testing, I will not be able to follow the waterfall model effectively. Instead I will be using Agile Development. This will allow my project to be more *exploratory*, experimenting with different means of DEM Generation, and testing methods as I implement them. See my Agile plan in the next section.

10. Project plan

Agile Plan

These User Stories will be updated and reordered during the project. They will be completed around my research, as a means of testing my findings and exploring their potential. This means some of these ideas are likely to be scrapped, or expanded, during the course of the project.

User story	Story points	Details	Importance	Rank
Prototype UI Design		No functionality, functionality will be added in future stories. Requires designing the UI, and implementing the UI in the development environment.	Required	10
Track devices GPS and store it locally		Should use threading to track the devices location. Must have on/off function. Store this GPS data on the device.	Required	20
Script for generating height map		Create a script which accesses the stored GPS data, and generates a height map. Should store locally.	Required	30

3D model generation from height map		Implement the functionality to generate a 3D model from the height map data. May require a spike to find the best means to do this.	Required	40
Add ability to record specific runs, on top of the normal GPS tracking.		This should simply add meta data to the GPS data. E.G. If the user has hit the start button to record a specific run, all GPS data generated after that point should include a key stating that it is part of that run. This will allow the program to display specific runs on the 3D model.	Optional	80
Use the 'specific run' data to display these runs on the 3D model.			Optional	90
Communicate GPS data between devices		Will require research spike. Needs the ability to choose a whitelist of people the user wants to share the information with. Possibly need a user login system and a server to communicate.	Optional	50
Show friends previous and current locations			Optional	60
Performance tracking		Create feature that extrapolates the GPS data to display statistics such as average speed, altitude etc.	Optional	100

Gantt chart

	Task Name	Start	Finish
1	**Research**	11/01/17	11/23/17
2	Current similar applications	11/01/17	11/07/17
3	GPS Collection	11/08/17	11/17/17
4	3D Modelling of terrain	11/08/17	11/13/17
5	Terrain data models	11/13/17	11/17/17
6	Height-maps	11/13/17	11/17/17
7	Alternatives	11/13/17	11/17/17
8	GPS Accuracy	11/18/17	11/23/17
9	Improvement techniques	11/18/17	11/23/17
10	**Planning**	11/01/17	12/06/17
11	Requirements	11/01/17	11/15/17
12	Survey	11/01/17	11/08/17
13	Requirements write-up	11/09/17	11/15/17
14	Design	11/23/17	12/06/17
15	Methodologies	11/23/17	11/28/17
16	User Interface	11/28/17	12/06/17
17	Program design	11/28/17	12/06/17
18	**Implementation**	12/07/17	02/26/18
19	Complete User Stories, follow Agile process	12/07/17	02/26/18
20	Write Unit Tests	12/07/17	02/26/18
21	**Report Write-up**	11/01/17	04/27/18

11. Legal, ethical, professional, social issues

I will be conducting a data collection exercise. I will create a participant information sheet to explain that I will be collecting their GPS data during the exercise via a mobile application. I will inform them that it will only collect data once they have turned the collection functionality on manually, and that it can be turned off whenever they want. I will then get their signature of informed consent.

The GPS data collected from the participants will not be linked to them in any way. If I need to refer to a specific GPS data set, they will simply be labelled 'A', 'B', etc. The data will automatically be encrypted as soon as it is generated, ensuring that no 3rd party will have access to it.

As for the final product, the users GPS data will only be stored locally on their device, encrypted, and shared only with a whitelist of users of their choice.

I have completed an ethical examination checklist and consent form.

Signatures

	Signature:	Date:
Student		19 October 2017
Project supervisor	Jonathan Mark Crellin	19 October 2017

Certificate of Ethics Review

Ethical Examination certificate University of Portsmouth

Certificate of Ethics Review

Project Title:	Exploring the potential of Digital Elevation Model generation from GPS in consumer products.
User ID:	721426
Name:	Jaco van Cranenburgh
Application Date:	11/10/2017 20:38:09

You must download your referral certificate, print a copy and keep it as a record of this review.

You should **submit your certificate to your FEC representative for further review**.

The FEC representative for the School of Computing is Carl Adams

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

All projects involving human participants need to offer sufficient information to potential participants to enable them to make a decision. Template participant information sheets are available from the:

- Univeristy's Ethics Site (Participant information template).

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

SchoolOrDepartment: SOC

PrimaryRole: UndergraduateStudent

SupervisorName: Jonathan Crellin

HumanParticipants: Yes

ParticipationLimitedToAnsweringQuestionsOrInterviews: No

HumanParticipantsWarning

ParticipantInformationSheets: I will be conducting a data collection exercise. I will create a participant information sheet to explain that I will be collecting their GPS data during the exercise via a mobile application. I will inform them that it will only collect data once they have turned the collection functionality on manually, and that it can be turned off whenever they want. I will then get their signature of informed consent.

ParticipantConfidentiality: The GPS data collected from the participants will not be linked to them in any way. If I need to refer to a specific GPS data set, they will simply be labelled 'A', 'B', etc.

Certificate Code: 4DEF-BC02-8068-580F-2987-098E-A068-A856 Page 1
InvolvesNHSPatientsOrStaff: No

NoConsentOrDeception: No

CollectingOrAnalysingPersonalInfoWithoutConsent: No

InvolvesUninformedOrDependents: No

DrugsPlacebosOrOtherSubstances: No

BloodOrTissueSamples: No

PainOrMildDiscomfort: No

PsychologicalStressOrAnxiety: No

ProlongedOrRepetitiveTesting: No

FinancialInducements: No

PhysicalEcologicalDamage: No

HistoricalOrCulturalDamage: No
InvolvesAnimals: No

HarmfulToThirdParties: No

OutputsPotentiallyAdaptedAndMisused: No

HasSecurityImplications: Yes

SecurityImplicationsDescription: Anonymity of the participants, and their position could be exposed if accessed by a 3rd party.

MinimizingSecurityRisks: The application will automatically encrypt the GPS data as soon as it is generated, ensuring that no 3rd party will have access to it.

Confirmation-ConsideredDataUse: Confirmed

Confirmation-ConsideredImpactAndMitigationOfPontentialMisuse: Confirmed

Confirmation-ActingEthicallyAndHonestly: Confirmed

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor signature: Jonathan Mark Collin

Date: 19 October 2017

Review by FEC Representative

The supervisor must have reviewed and signed above before this section can be completed.

☐ Synopsis and supporting information has been provided.

Name of representative:

Comments:

Representative signature:

Date:

Certificate Code: 4DEF-BC02-8068-580F-2987-098E-A068-A856 Page 2

Survey Questionnaire

Sports mobile application survey.

I am a Computer Science student doing my final year project. If you have time, please answer this survey. All answers are anonymous.

If you supply an email address at the end, you have the chance to win a pair of hand-made bamboo sunglasses made by Bambooka.org!

*Required

1. **Age** *

___

2. **Sex** *

Mark only one oval.

- Female
- Male
- Prefer not to say
- Other: ___

3. **What phone do you have? Tick all that apply** *

Tick all that apply.

- [ ] Android
- [ ] iPhone
- [ ] Windows Phone
- [ ] I don't know
- [ ] Other: ___

4. **How frequently would you say you use mobile apps?** *

Mark only one oval.

- Very frequently
- Somewhat frequently
- Occasionally
- Rarely
- Not at all

5. **Do you currently use any sports software or applications?** *

Mark only one oval.

- Yes
- No

6. **If yes, what benefits do you ike most about the app/s? Tick all that apply.**
   Tic all that apply.

- [ ] Improvement - Performance tracking and statistics
- [ ] Ease of use
- [ ] Fun - Entertainment value
- [ ] Motivation
- [ ] Safety - Location sharing for emergencies
- [ ] Other: ______________________

7. **When using a sports app, rank the importance of the following benefits:** *
   Mark only one oval per row.

	1 - Most important	2	3	4 - Least important
Entertainment - the app provides a source of fun	◯	◯	◯	◯
Performance - Track and improve your performace	◯	◯	◯	◯
Safety - the app makes sports activity more safe	◯	◯	◯	◯
Health - enables you to be active and healthy	◯	◯	◯	◯

8. **When using a sports app, rank the importance of the following features:** *
   Mark only one oval per row.

	1 - Most important	2	3	4	5 - Least important
Accuracy - the GPS data should be as accurate as possible	◯	◯	◯	◯	◯
Visual style/appeal - the app should look good	◯	◯	◯	◯	◯
Statistical Insight - should provide you with relevant data	◯	◯	◯	◯	◯
Ease of use	◯	◯	◯	◯	◯
Connectivity to others/social media	◯	◯	◯	◯	◯

I am creating a snow sports (skiing/snowboarding) mobile app. Please read the description below and answer the following questions.

The aim is to create a snow sports mobile application that utilizes GPS. My application will use the GPS data that your phone collects while skiing or snowboarding, and generate a 3D model of the mountain. The idea is that the 3D model will get larger and more accurate the more you explore, whilst spending a day on the slopes.

9. **How appealing would you find this app?** *
   Mark only one oval.

	1	2	3	4	5	
Very appealing - likely to use						Not appealing - will not use

10 **How credible do you think th s app is? Do you think this app would work as described?** *
Mar   only one oval.

	1	2	3	4	5	
Very credible						Not credible

11. **How different is this app idea to what is already out there?** *
Mark only one oval.

	1	2	3	4	5	
Very unique						Not unique

12. **How likely are you to use this app, assuming you were to go skiing/snowboarding?** *
Mark only one oval.

	1	2	3	4	5	
Very likely						Not likely

13. **If you would like to expand on any of your answers, please do so here.**

14. **What other features would you find useful in a snow sports mobile application? Tick all that apply** *
Tick all that apply.

- [ ] View friends locations on the model to make it easier to meet up
- [ ] Record specific runs, compare speeds with friends or try beat personal best
- [ ] Performance tracking, display statistics
- [ ] Link with external devices such as Fitbit, track bio-metrics and other extra information
- [ ] Other:

15. **Do you do any snow sports? Tick all that apply.** *
Tick all that apply.

- [ ] Ski
- [ ] Snowboard
- [ ] None
- [ ] Other:

16 **How would you rate your skill level?** *

Mar only one oval.

- Beginner
- Intermediate
- Advanced
- Professional
- Not applicable

17. **If you would like to be entered to win a pair of homemade bamboo sunglasses, please provide your email address below.**

____

18. **Would you like to test the prototype of this app in the future, and are okay with being contacted?**

Mark only one oval.

- I give permission to contact me
- No, thanks

Questionnaire Results

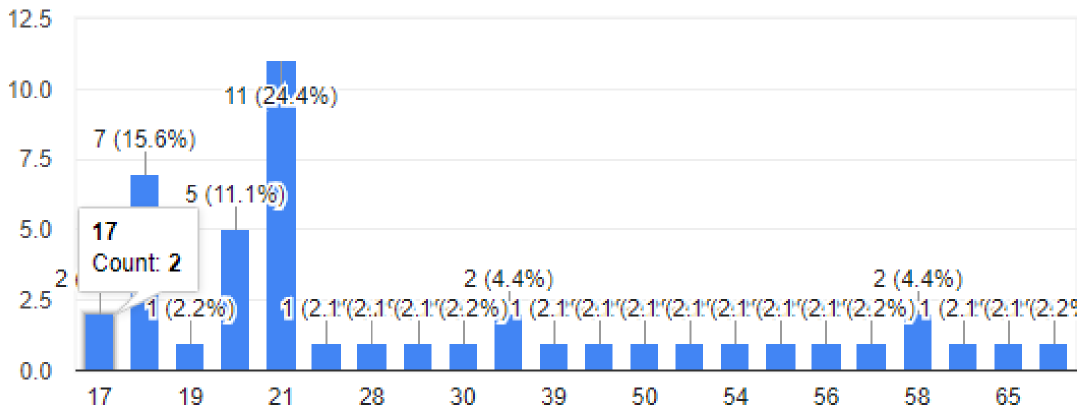
Sports mobile application survey.

45 responses

Publish analytics

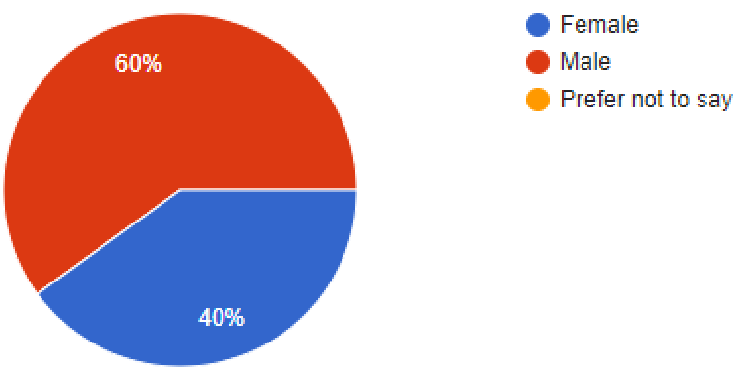
Age

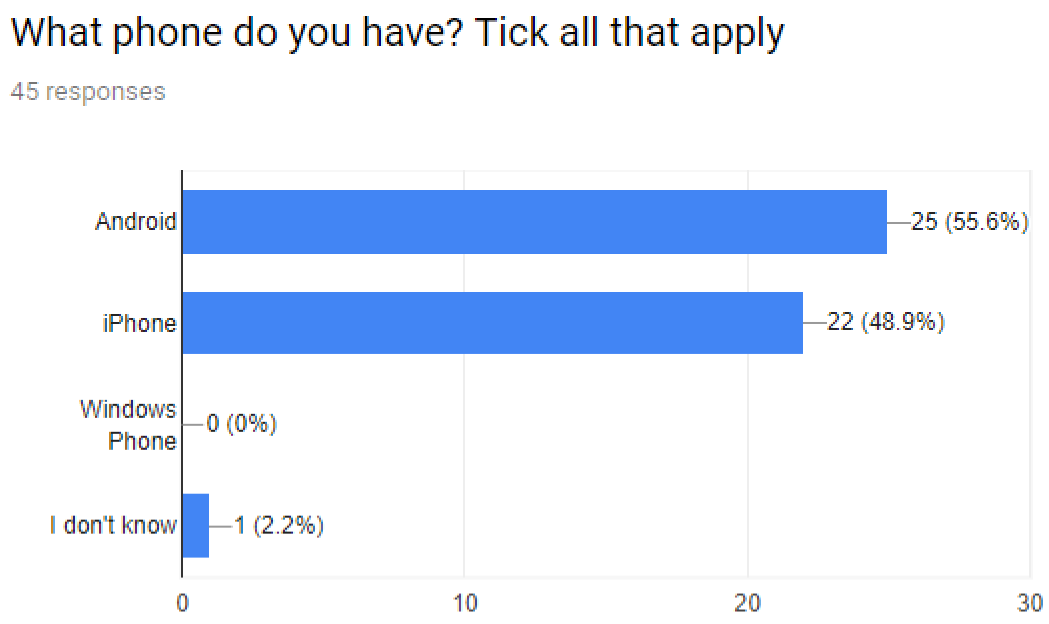
45 responses

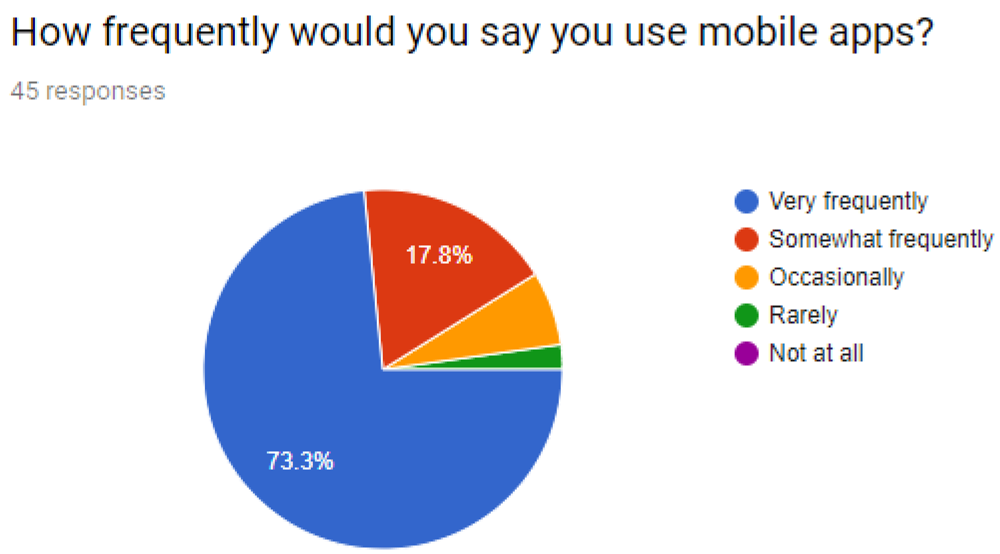


Sex

45 responses

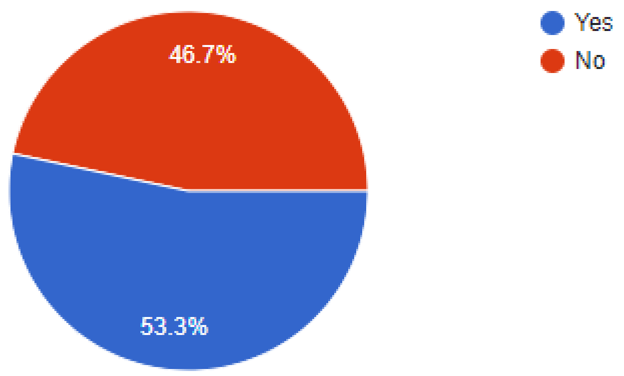



What phone do you have? Tick all that apply
45 responses
Android
25 (55.6%)
iPhone
22 (48.9%)
Windows Phone
0 (0%)
I don't know
1 (2.2%)
0
10
20
30


How frequently would you say you use mobile apps?
45 responses
17.8%
73.3%
Very frequently
Somewhat frequently
Occasionally
Rarely
Not at all

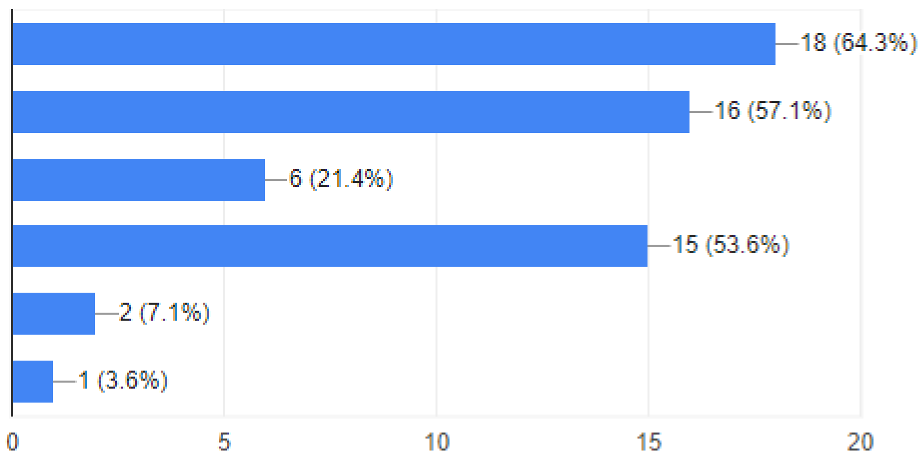
Do you currently use any sports software or applications?

45 responses

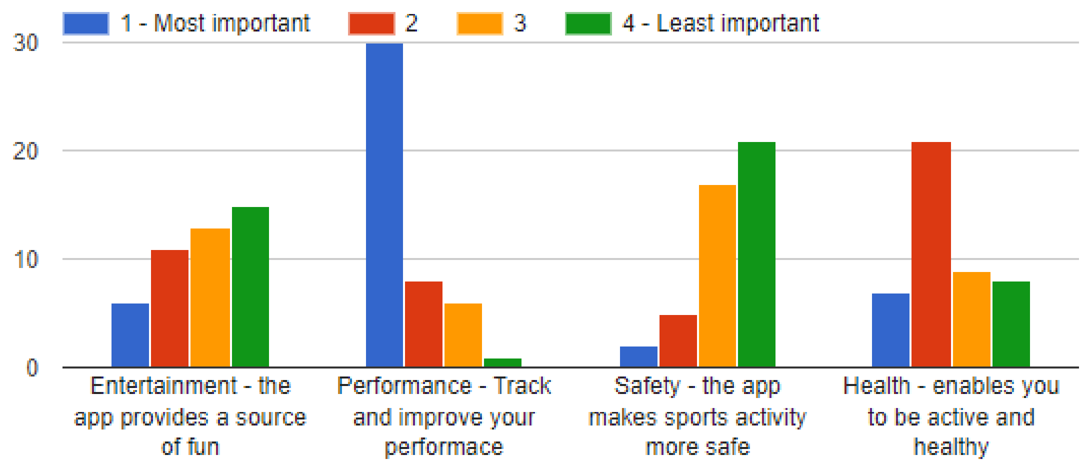


If yes, what benefits do you like most about the app/s? Tick all that apply.

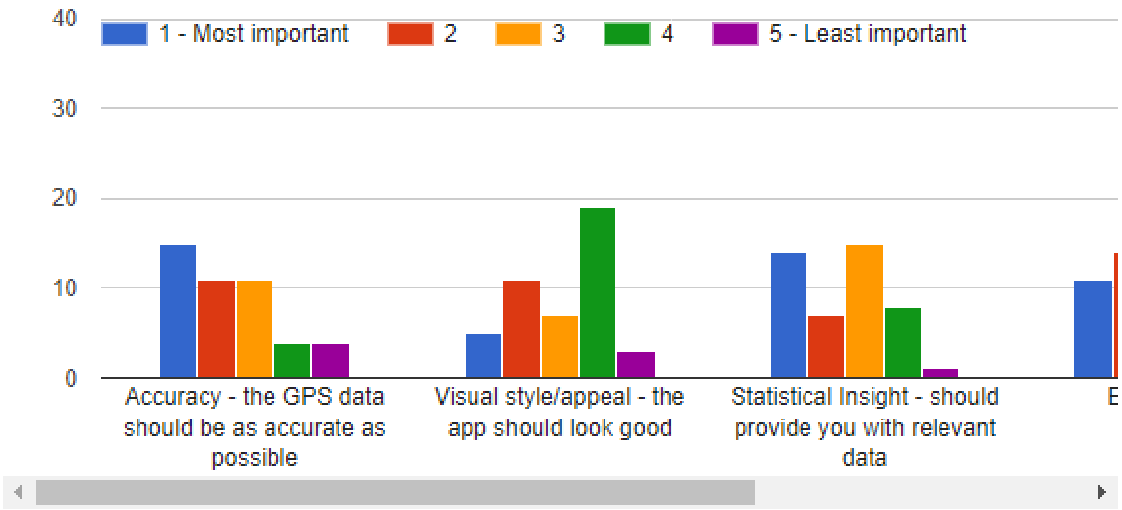
28 responses



When using a sports app, rank the importance of the following benefits:

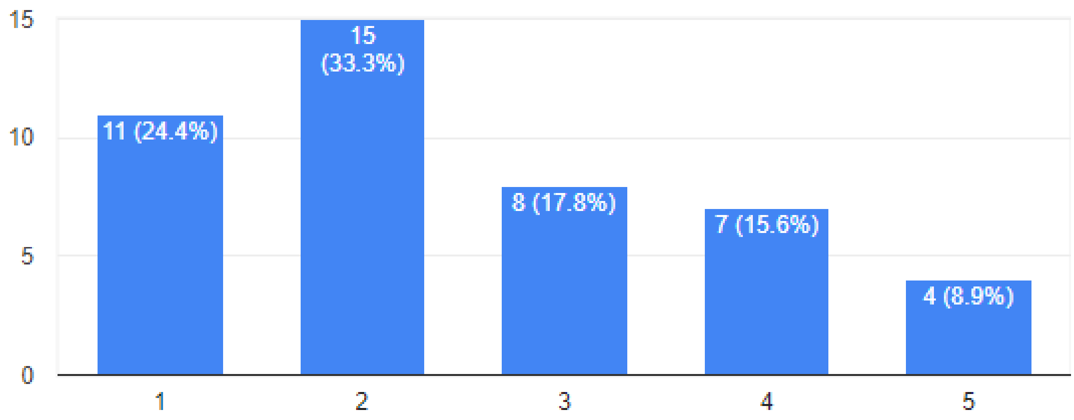


When using a sports app, rank the importance of the following features:



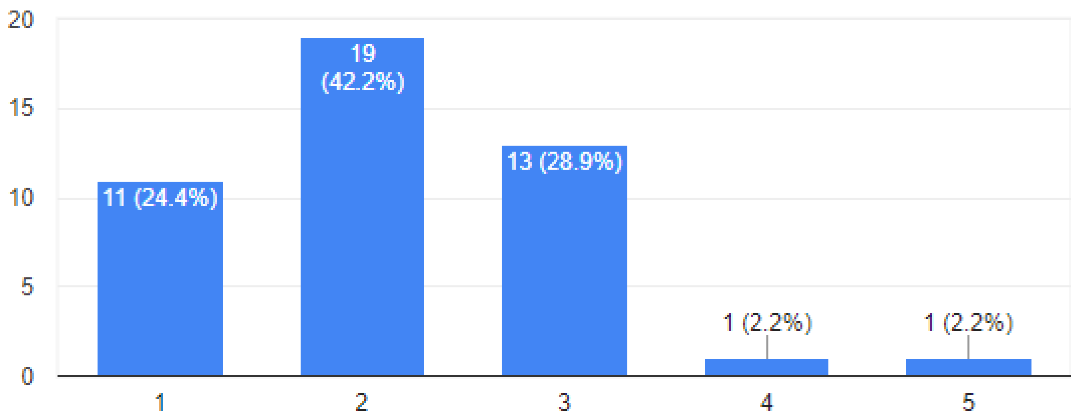
How appealing would you find this app?

45 responses



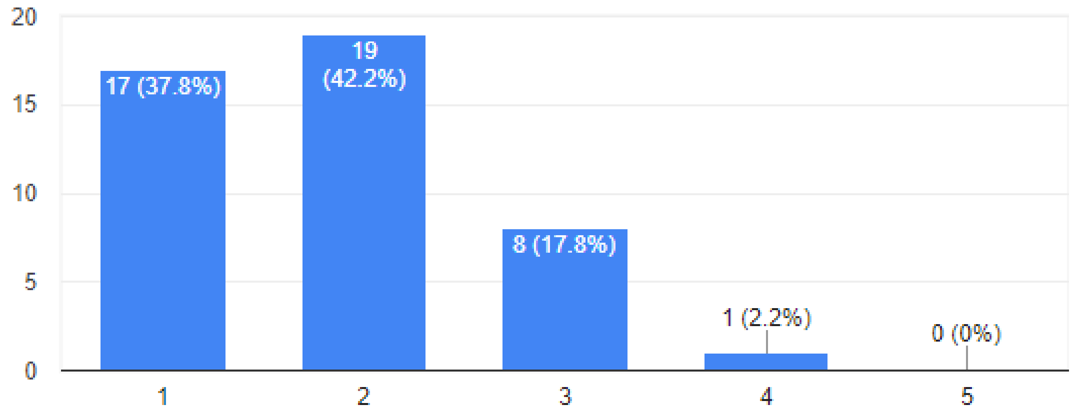
How credible do you think this app is? Do you think this app would work as described?

45 responses



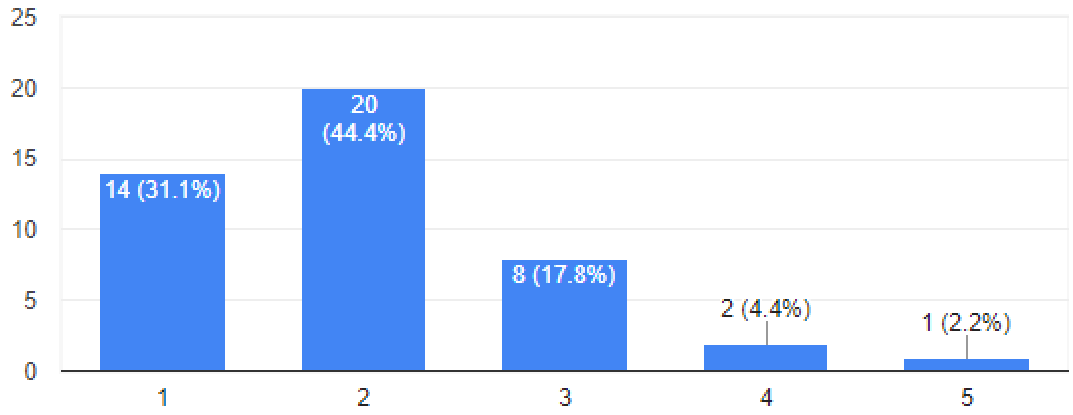
How different is this app idea to what is already out there?

45 responses



How likely are you to use this app, assuming you were to go skiing/snowboarding?

45 responses



If you would like to expand on any of your answers, please do so here.

8 responses

nah i'm good

the app collects data from other users (on same mountain/slopes) .. incase users only use one single slope? people rate slopes? comment them? .. app suggests for you? .. build a better image of mountain from other users? separate from your own '3D image'? ... would also show routes as well no? .. soz rambles .... scrolls down.. OHH

How do you use the ap whilst skiing - use one stick?

I'd be interested in what someone would do with a 3D model of a mountain just skied. How different is this to Ski Tracks or Realski (USA)? What problem is being solved here?

Question When using a sports app, rank the importance of the following benefits: * This question and the one that follows it, doesn't show properly unless screen is in landscape.

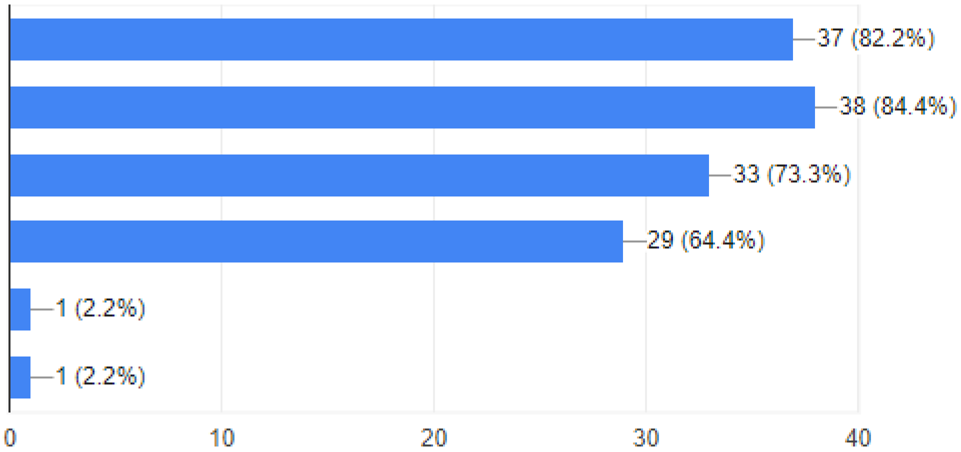
Looks unique and interesting, however wouldn't be sure how many times I would use it since there isn't necessarily rewarding for the user of the app (maybe like Pokemon GO). Unless you're able to combine data from many users, it will take a very long time for one person to collect enough data to form the mountain.

I would use it if it also had other features - unlocking features as you go or using the map as a functional map, last ski lift times if you're far away from the resort

It assumes the skier will explore the mountain, not just several courses. The relevance of the 3D map is only important to serious skiers, not just hobby skiers that will ski for a few hours

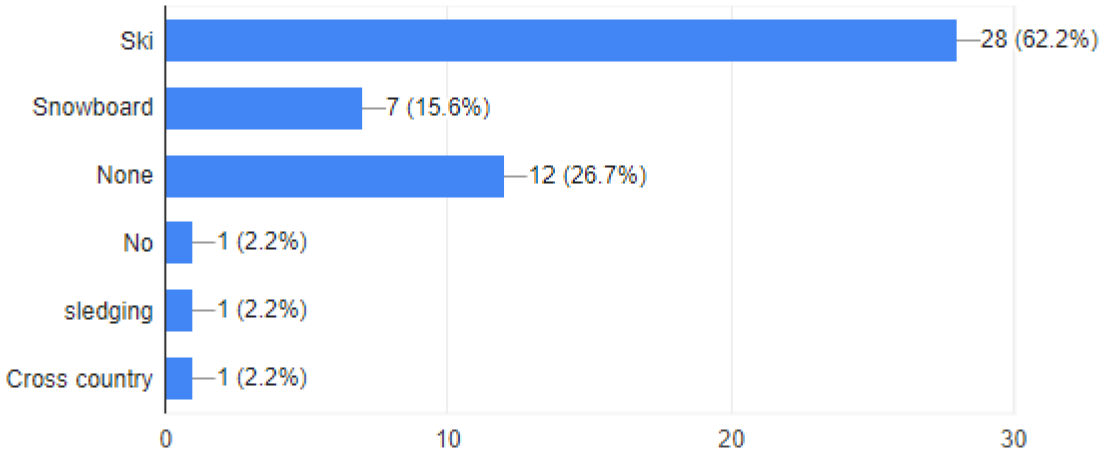
What other features would you find useful in a snow sports mobile application? Tick all that apply

45 responses



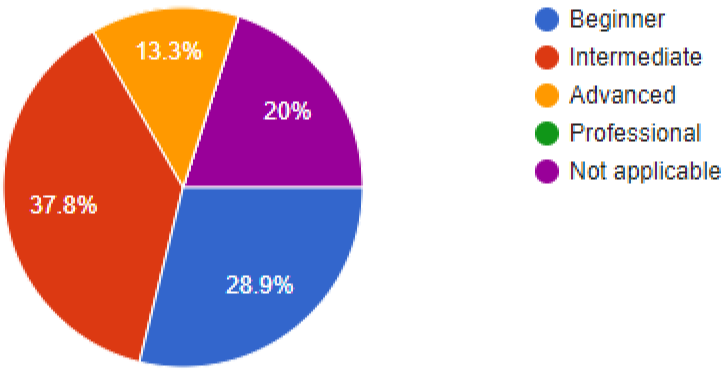
Do you do any snow sports? Tick all that apply.

45 responses



How would you rate your skill level?

45 responses



Would you like to test the prototype of this app in the future, and are okay with being contacted?

45 responses