John Knox

Socially Enhancing Track Based Sports

Project Proposal

Churchill College

October 19, 2012

Proforma

Name:	**John Knox**
College:	**Churchill College**
CRS-ID:	**JK510**
Project Title:	**Socially Enhancing Track Based Sports**
Project Originator:	Dr John Fawcett
Supervisor:	Dr John Fawcett

Introduction and Description of the Work

This project implements an Android app for use with track based sports.

The app presents a map of the track on screen with the other participants plotted at their corresponding positions, similar to the Heads Up Display on racing video games. The app continuously sends data from each device to each of the others, aiming for sufficient frequency to provide an accurate display at all times. Due to the nature of some track-based sports such as mountain biking, mobile signal strength may be intermittent so the protocol designs will have to expect heavy packet loss. The project is an investigation and evaluation of the effectiveness of different communication protocols in delivering near real-time updates despite the varying and at times adverse network characteristics. Comparison metrics include frequency of position updates, network data usage, and update delay, as well as how the solutions scale with the number of connected devices.

Resources Required

- My Laptop Computer

  Intel Core i3 M380 @ 2.53GHz

  4GB RAM

  Windows 8 running VirtualBox with Linux Mint

- My Android Phone

  ZTE Blade

Starting Point

Part 1B Networking course taken in previous year.

Substance and Structure of the Project

The project will develop the functionality of a basic Android app to visualise the device's knowledge of the current and historic positions of each participant and to record logs for evaluation. The interface created will be a tool for this scientific evaluation; there is no requirement to deliver a polished, production-ready graphical user interface for ordinary users. Nonetheless, this development requires me to become familiar with the Android system, development environment and build/compile tool chain, and time has been allocated for this.

I will design a modular piece of software for the app, leaving a gap for the networking implementation. This will allow me to drop in different implementations of a network module and compare networking approaches and protocols.

Included in the basic visualisation functionality is the capability to draw the track to the screen, and plot and update player positions. The app must use the Android location APIs to turn on the GPS and record its position at regular intervals. The app must drive the various communications devices – WiFi, mobile internet, bluetooth – as required by the network protocol modules.

The motivating use case means there must be a mechanism to define the group of phones that will work together for one track sport session. This project will investigate the networking and security issues in this area and develop a session set-up "screen" for the app. Once the participant group is established, there should be no cross-talk between independent groups and due consideration for data privacy is included. The evaluation will reflect on the security and privacy properties delivered by the session set-up protocol(s).

Once the track sport is underway, further networking issues become interesting. The approaches taken by the alternative communications modules that I will develop will be divided into client-server, and peer-to-peer ("p2p") based methods. The centralised protocols will use a server application built in Java as the coordinator between devices.

The project includes some time to research network protocols, investigating how other applications use the network connectivity and (for example) how Skype tolerates packet loss. The protocols' goals are frequent and up-to-date positions being displayed in near real time. The output of that research will be designs for the network protocols that I will implement and evaluate – at least one client-server protocol and at least one p2p.

The p2p solution will provide the same functionality as client-server but without the need for a server, using mobile internet, wifi, or potentially other channels if possible. However, it is accepted that mobile operators commonly use Network Address Translation and block incoming connections, so in practice a truly p2p protocol operating over mobile internet may not be compatible with most mobile networks in the real world. Despite this, it may provide valuable results to compare against the server based methods and is highly applicable to other networks: private mobile radios, ZigBee, bluetooth etc.

To avoid the need for multiple Android devices, with their inevitable nuances, I will develop a PC client to simulate many devices running the app in order to see how the protocols scale.

I will also spend a large amount of time evaluating the implementations I make, which will involve adding functions to log data and later analysing it to compare the performance of the different approaches.

The project will be composed of the following sections:

- Familiarising myself with the Android framework, libraries, tool chain and development environments.
- Making a basic app and designing and implementing the functionality for displaying location data and for driving the GPS and communications APIs.
- Designing and implementing a secure session set-up procedure.
- Researching network protocols.
- Implementing alternate communication modules, and the server application.
- Testing and evaluation of network modules.
- Writing dissertation.

Possible extensions of the project

The app could be extended to add voice chat support, so that short audio messages could be unicast, multicast or broadcast to the other participants. This extension would place different demands on the network modules and hence motivate the use of different protocol-layer techniques.

A ghost replay function could be added so that users can race against their (or other's) previous performances around the track.

More functionality could be added to the app so that it does further processing on the data, such as producing graphs of speed and other statistics, using local GPS and data from the other participants' records.

A professionally designed user interface could be added and evaluated (using user trials).

Success Criteria

The following should be achieved:

- App displays an on-screen visualisation of data from the local GPS.
- App plots positions of up to at least four participants (including itself) in near real-time on the map.
- At least one server based method and one peer to peer method of communication has been implemented, with comparisons of their performance made.

Timetable

The following is my plan for the project.

"Additional" in this context means that this task is desirable but not necessary so may be dropped if time is short without impacting the core deliverables of the project.

Weeks 1,2 [20th Oct 2012 - 2nd Nov 2012]

Design the modular software structure for the Android app. Draw up initial UML class diagrams and list the required 'screens'.

Learn to use the Android framework, install Android SDK and get a "hello world" app working.

Begin iteratively defining the "hello world" app into the project app. This work continues in the next stage.

Milestone, 2 Nov: UML class diagrams agreed with supervisor; "hello world" app running on an Android phone.

Weeks 3,4 [3rd Nov - 16th Nov 2012]

Finish implementing the basic app from weeks 1,2, including developing code to draw the positions and trails of each participant on the screen.

Develop code to turn on the Android device's GPS and interpret the data to draw trails on the map. Kick-the-tyres test by walking around the college fields and checking the trace resembles the path walked.

Additional: Print real time data and historical statistics (in numeric format) on the device (in numeric format, for later debugging).

Milestone, 16 Nov: Application has all tracking and displaying functionality, with crude testing performed.

Week 5,6 [17th Nov - 30th Nov 2012]

Research and write up possible communication protocols, including how their performance (in the context of this application) can be meaningfully analysed. Write up and meet with supervisor to identify which protocols will be implemented and how their evaluation will work. This research will be used in weeks 7,8,9,10 as well as when writing the final dissertation.

Milestone, 30 Nov: Written up research results submitted to supervisor, and agreement with supervisor on which protocols I will implement.

End of Michaelmas term [30th November 2012]

Week 7 [1st Dec - 7th Dec 2012]

Research and implement a solution for setting up a secure session with other devices. Use knowledge gained from research in weeks 5,6 to decide what information will have to be exchanged to set up the session.

Additional: Implement another session set up method so ease of use, simplicity and security can be compared.

Milestone, 7 Dec: App now complete except for networking module.

Weeks 8,9 [8th Dec - 21st Dec 2012]

Implement the protocols chosen in weeks 5,6 using the session as set up by the work done in week 7.

The implementations will be in the form of modules that will slot into the rest of the application framework interchangeably, so they can be easily compared later on in the project. This will involve making the server application.

This stage will be continued after a 1 week Christmas break.

Week 10 is contingency – the implementation of these protocols is expected to take three weeks, but an extra is included to absorb any slip at this stage in the project, so testing and evaluation can begin on schedule in January.

Milestone, 21 Dec: Now at least one complete implementation of the app.

Christmas week [22nd Dec 2012 - 28th Dec 2012]

No work is scheduled for this holiday period.

Week 11 [29th Dec 2012 - 4th Jan 2013]

Return to finish the work on implementing protocols from the previous stage. Milestone, 4 Jan: App now complete with several different interchangeable communications modules, as well as a server application to serve the android devices.

Weeks 12,13,14 [5th Jan 2012 - 25th Jan 2013]

Carry out full system tests. This is expected to include some unit testing of network protocol state machines and some live functional verification, to be determined (and agreed with supervisor) in light of implementing the code and better understanding where the risks/bugs lie.

Add to the framework so that run-time statistics are logged automatically.

Evaluate the network implementations made in weeks 8,9. Uses knowledge of which parameters/metrics to use from weeks 5,6.

Analyse the runtime logs to produce results that I will use to compare the performance of the network modules.

Prepare progress report for deadline [1st Feb 2013] and presentation for event [7th Feb 2013]

Milestone, 18 Jan: Have informally written up results from the evaluation tests, that can be used to compare the different implementations in the next stage.

Beginning of Lent term [15th Jan 2013]

Week 15,16 [26th Jan 2013 - 8th Feb 2013]

Write introduction chapter and section headings/bullet points for the remaining chapters.

Milestone, 9 Feb: Introduction and section headings with supervisor for review.

Week 17,18 [9th Feb 2013 - 22nd Feb 2013]

Write preparation and implementation chapters.

Milestone, 23 Feb: These chapters submitted to supervisor for review.

Week 19,20 [23rd Feb 2013 - 9th Mar 2013]

Write Evaluation and Conclusion chapters.

Milestone, 9 Mar: These chapters submitted to supervisor for review.

Week 21 [9th Feb 2013 - 16th Mar 2013]

Incorporate review feedback from supervisor.

Milestone, 16 Mar: First complete draft of dissertation submitted to supervisor for review.

End of Lent term [15th March 2013]

Easter vacation [16th March 2013 - 22nd April 2013]

Extensions and additions to the dissertation, as time permits.

Exam revision.

Final deadline: hand in dissertation by noon on 17th May 2013

Version Control and Back-Up

I will be doing all of the main work on my laptop. I will use Git with a remote repository on the PWF system, where updates will be periodically pushed to, as well as Dropbox backing up the project continuously. In the event that my own computer fails I will use the PWF machines to continue.