The Fragments Route Plotter App

The following files form part of the Android app:

Java files:
- MainActivity.java
- SpeedoFragment.java
- ListFragment.java
- GpsStamp.java
- StampAdapter.java

XML files:
- AndroidManifest.xml
- activity_main.xml
- listitem.xml
- listfragment.xml
- speedofragment.xml
- strings.xml

Two screenshots of the app running on a tablet, one on Portrait and the other on Landscape.

The layout in this app comprises of two fragments – one to contain the "speedometer" element, i.e., displaying the speed, time and GPS coordinates, and the other to contain the "list" element, which displays the last few readings.

Hence the word "Fragment" in the app's name.

The layout is structured into fragments to aid adaptation to each orientation – the fragments will appear side-by-side in Landscape, and one above the other in Portrait.

The classes

The following are the classes, and what they do:

MainActivity: - this is where the app starts. It is also the class through which the fragment classes talk to each other – fragment classes should never talk to each other directly.

It controls the frequency of GPS readings, and what is done when a GPS signal is read – it creates a new GpsStamp object, and tells the SpeedoFragment to display the values. If the Recording switch is on, the GpsStamp object is added to an array that is created when the app starts, and this array is updated on the ListFragment.

When the number of GpsStamp objects in the array reaches 12, a '.TXT' file is opened and these values are appended to the file.

When the app is closed down, all the GpsStamp values in the entire '.TXT' file are read, and written into a JSON file.

SpeedoFragment: - this is the class that control what goes on with the text boxes to display the current speed, time and GPS coordinates.

ListFragment: - this class controls the listing of the most recent readings.

GpsStamp: - this class is that contains values for GPS coordinates at a certain point in time – speed (in metres per second), longitude, latitude and time (in milliseconds since 1/1/1970). It is in this class that the speed value is converted to kilometres per hour, and time is converted to a more "human readable" format.

StampAdapter: - this class manages the data in the GpsStamp objects, and adapts it to the individual items in the ListFragment.

The Android Manifest

The only additions that had to be made to the "boilerplate code" in this XML file are:

- to grant permission to access GPS and internet settings
- to lock the screen orientation. This means that whatever the screen orientation is when the app is started, that is orientation is stays at, and does not change when the screen is rotated.

The HTML and associated files

The following files were created, to demonstrate the workings of the app.

- DublinToCork.json
- CorkToDublin.json
- map_route_plotter.html
- map_route_plotter.js
- style.css

The two JSON files were created by the Android app, just before I shut it down, while I was taking a trip by train from Cork to Dublin, and back one Saturday. On the trip up to Dublin, the train stopped at the stations in Mallow, Limerick Junction, Thurles and Portlaoise. And on the way back, we stopped at Thurles, Limerick Junction, Charleville and Mallow.

The HTML file demonstrates how the route that was plotted by the app, is drawn onto a map. Here, I am using OpenLayers, an open source alternative to Google Maps. The line shown in the example indicates the route taken by the train, and also the train's speed. There is a key to the left of the screen, to indicate the speed range for the different colours.

I recommend using MS Visual Studio, to view this file. It's free. In order to be able to view the HTML, you need to get Live Server extension to Visual Studio.

A few questions you will probably be asking about the map when you view it are: How come we have a dead straight line just outside Cork Station? This is because there is a tunnel next to the station, taking the tracks through a hill to get to the north of the city, and we could not get a good GPS signal in there. On the way back, we seemed to be going more slowly in a lot of places between Dublin and Thurles. This is because there were several speed restrictions, which the driver explained for the later-than-advertised arrival into Cork. I suspected two possible reasons: engineering works on the track, and traffic.

Anyway, guys, feel free to download this code, play around with it. Although I do recommend saving copies of the files before modifying them, that way, if worst comes to worst, you will be able to fall back on the as-downloaded code.