

Software Engineering Large Practical: Location services—concepts and programming

Stephen Gilmore
(`Stephen.Gilmore@ed.ac.uk`)
School of Informatics

October 5, 2016

Contents

- ▶ Android software development
- ▶ Location services
- ▶ Testing location-aware apps

Android software development

- ▶ Android software development is supported by well-documented software APIs and software development platforms (such as Android Studio).
- ▶ It is also supported by many good tutorials with Android code snippets and example Android projects showing how APIs are used.
- ▶ **In this practical, you are encouraged to make use of example code which you find available in tutorials and Android code samples, and to include libraries as needed.**
- ▶ This is *re-use*, which is a good thing, not *plagiarism*, which is a bad thing.
- ▶ Please cite the sources which you used in developing your app.

The nature of software development

- ▶ The practice of software development has changed markedly over the last five to ten years.
- ▶ The existence of sites such as `stackoverflow.com` has made accessing relevant experience in application development much easier.
- ▶ The existence of open-source repositories such as GitHub has made accessing working example software projects much easier.
- ▶ The fact that both of these archives are searchable by Google makes them more accessible again.
- ▶ For Android specifically, the existence of `github.com/googlesamples/` provides many high-quality examples of Android projects.

Android software development in practice

1. Investigate relevant Android concepts using tutorials and documentation from developer.android.com/training/
2. Investigate code samples which provide examples of these concepts in use. Download and try these.
3. Identify relevant libraries and services to import into your project. Install these.
4. Add code to your project based on the concepts learned and example code seen, modifying as necessary.

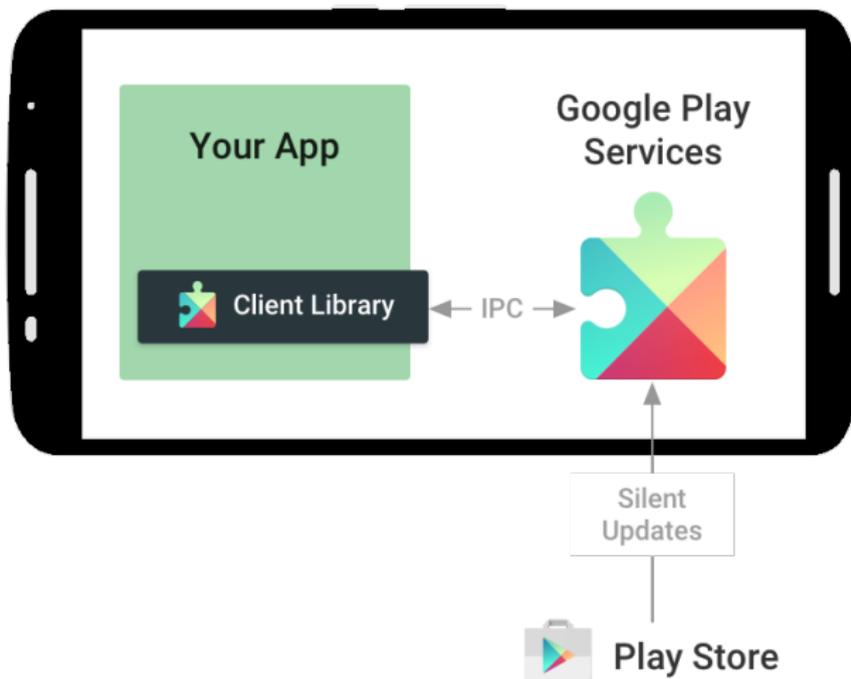


This contrasts with typical Java development where supporting materials for developers sometimes ends with API documentation. However, because Android is developing rapidly, not all examples are up-to-date (e.g. some use Maven instead of Gradle as their build system).

An example: Android location services

- ▶ Location-awareness is a core feature of apps and services for mobile devices. Services of all kinds can be enhanced with location-awareness (e.g. a search app providing the option to “find restaurants *near me*”).
- ▶ The *Google Play Services* location APIs in the package `com.google.android.gms.location` are the preferred way of adding location awareness to your app.
- ▶ *Google Play* services have a distinguished status within Android apps because they can be updated directly from Google Play (Google’s “app store”) and are invoked by inter-process communication from a client library in your app.

Google Play Services



Getting permission to access locations

- ▶ Apps that use location services must request location permissions using either `ACCESS_COARSE_LOCATION` or `ACCESS_FINE_LOCATION`.
- ▶ Permission is requested with the `uses-permission` element in your app manifest (`AndroidManifest.xml`).

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
2     package="com.google.android.gms.location.sample.basiclocationsample"  
3     >  
4     <uses-permission android:name="android.permission.  
5     ACCESS_FINE_LOCATION" />  
</manifest>
```

Getting permission to access locations

- ▶ Apps that use location services must request location permissions using either `ACCESS_COARSE_LOCATION` or `ACCESS_FINE_LOCATION`.
- ▶ Permission is requested with the `uses-permission` element in your app manifest (`AndroidManifest.xml`).

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
2     package="com.google.android.gms.location.sample.basiclocationsample"  
3     >  
4     <uses-permission android:name="android.permission.  
5     ACCESS_FINE_LOCATION" />  
</manifest>
```

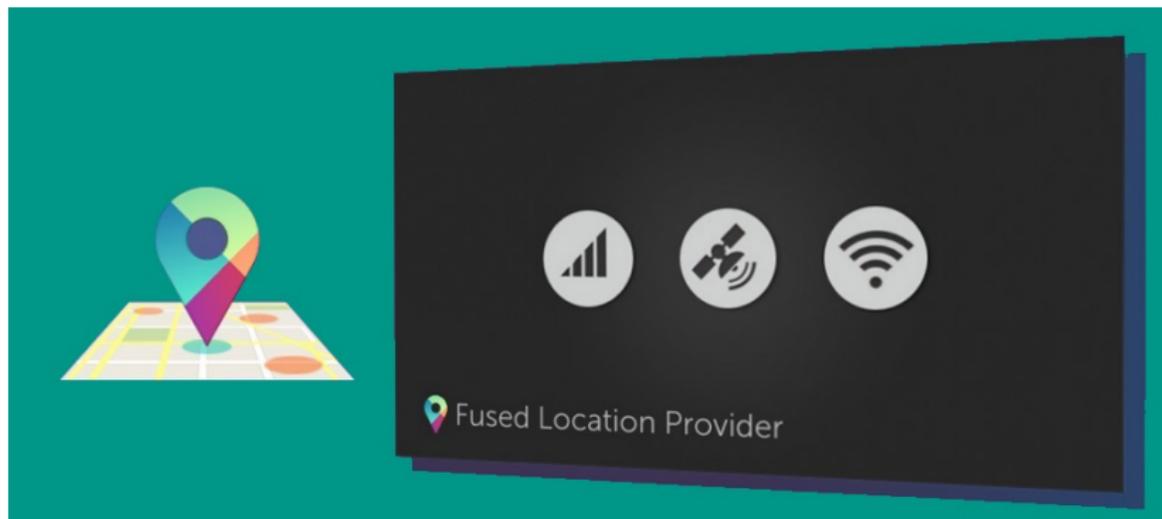
Question

Why are permissions requested in XML documents, rather than in Java code?

Retrieving the current location

- ▶ Using the Google Play services location APIs, your app can request the last known location of the user's device.
- ▶ Google Play services are part of *Google Mobile Services* (GMS).
- ▶ You will need to make use of
 - ▶ `com.google.android.gms.common.api`
 - ▶ `.GoogleApiClient` (abstract class)
 - ▶ `com.google.android.gms.location`
 - ▶ `.FusedLocationProviderApi` (interface)

Fused Location Provider



The *Fused location provider API* automatically chooses the best location provider which is suited for your hardware of your Android device.

Activity onCreate()

```
1 // Create an instance of GoogleApiClient.
2 if (mGoogleApiClient == null) {
3     mGoogleApiClient = new GoogleApiClient.Builder(this)
4         .addConnectionCallbacks(this)
5         .addOnConnectionFailedListener(this)
6         .addApi(LocationServices.API)
7         .build();
8 }
```

Activity onStart() and onStop()

```
1 protected void onStart() {  
2     mGoogleApiClient.connect();  
3     super.onStart();  
4 }  
5  
6 protected void onStop() {  
7     mGoogleApiClient.disconnect();  
8     super.onStop();  
9 }
```

Get the last known location of a device

```
1 public class MainActivity extends ActionBarActivity implements
2     ConnectionCallbacks, OnConnectionFailedListener {
3     ...
4     @Override
5     public void onConnected(Bundle connectionHint) {
6         mLastLocation = LocationServices.FusedLocationApi.getLastLocation(
7             mGoogleApiClient);
8         if (mLastLocation != null) {
9             mLatitudeText.setText(String.valueOf(mLastLocation.
10                 getLatitude()));
11             mLongitudeText.setText(String.valueOf(mLastLocation.
12                 getLongitude()));
13         }
14     }
15 }
```

Get the last known location of a device

```
1 public class MainActivity extends ActionBarActivity implements
2     ConnectionCallbacks, OnConnectionFailedListener {
3     ...
4     @Override
5     public void onConnected(Bundle connectionHint) {
6         mLastLocation = LocationServices.FusedLocationApi.getLastLocation(
7             mGoogleApiClient);
8         if (mLastLocation != null) {
9             mLatitudeText.setText(String.valueOf(mLastLocation.
10                getLatitude()));
11             mLongitudeText.setText(String.valueOf(mLastLocation.
12                getLongitude()));
13         }
14     }
15 }
```

Question

Why might `getLastLocation()` return null?

Setting up a location request

- ▶ Locations can be used for different purposes in different apps, and some require greater accuracy than others.
- ▶ Location services consume energy and can drain a device's battery quickly, so there is a trade-off to be made between high accuracy and low energy.
- ▶ The `FusedLocationProviderApi` can be configured using a `LocationRequest` which sets the update interval, fastest update interval, and priority values for location requests.
 - ▶ `setInterval()` sets the rate in milliseconds at which your app would like to receive location updates
 - ▶ `setFastestInterval()` sets the *fastest* rate in milliseconds at which your app can handle location updates
 - ▶ `setPriority()` sets the priority of the request to `PRIORITY_BALANCED_POWER_ACCURACY`, `PRIORITY_HIGH_ACCURACY`, `PRIORITY_LOW_POWER` or `PRIORITY_NO_POWER`

Create the location request and setting the parameters

```
1 LocationRequest mLocationRequest = new LocationRequest();  
2 mLocationRequest.setInterval(10000); // preferably every 10 seconds  
3 mLocationRequest.setFastestInterval(5000); // at most every 5 seconds  
4 mLocationRequest.setPriority(LocationRequest.  
    PRIORITY_HIGH_ACCURACY);
```

Parameters such as these need to be tuned to find the best tradeoff between accuracy and energy efficiency for an app.

Requests and results

```
1 LocationSettingsRequest.Builder builder =
2     new LocationSettingsRequest.Builder ()
3     .addLocationRequest(mLocationRequest);
```

```
1 PendingResult<LocationSettingsResult> result =
2     LocationServices . SettingsApi . checkLocationSettings (
3         mGoogleClient,
4         builder . build ());
```

The final result object from a `PendingResult` can be retrieved:

- ▶ via blocking calls to `await()`, or `await(long, TimeUnit)`, or
- ▶ via a callback by passing in an object implementing interface `ResultCallback` to `setResultCallback(ResultCallback)`.

Changing Location Settings (1/3)

```
1 result . setResultCallback (new ResultCallback<LocationSettingsResult>()) {
2     @Override
3     public void onResult( LocationSettingsResult result ) {
4         final Status status = result . getStatus();
5         final LocationSettingsStates = result . getLocationSettingsStates
6             ();
7         switch ( status . getStatusCode()) {
8             case LocationSettingsStatusCodes.SUCCESS:
9                 // All location settings are satisfied . The client can
10                // initialize location requests here.
11                ...
12                break;
```

Changing Location Settings (2/3)

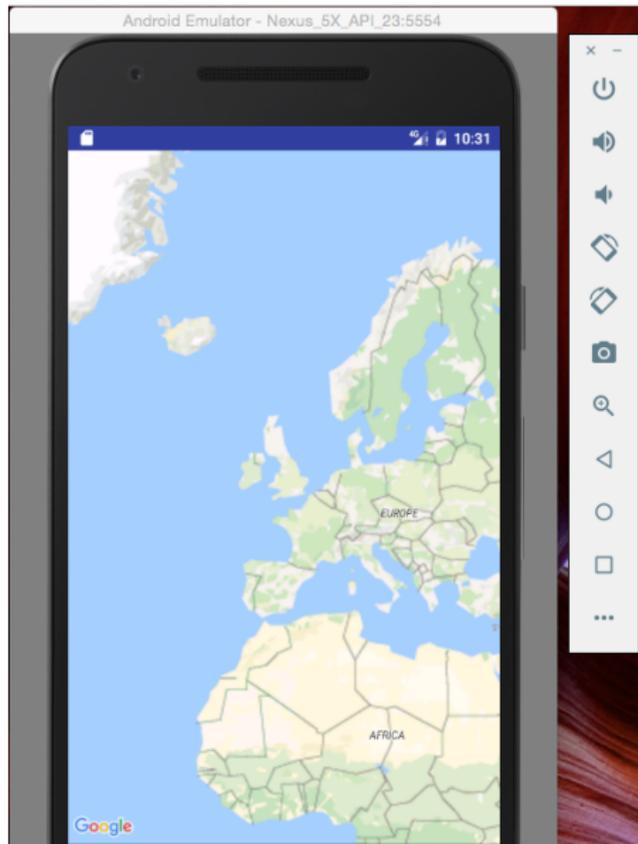
```
12     case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
13         // Location settings are not satisfied ,
14         // but this can be fixed
15         // by showing the user a dialog.
16         try {
17             // Show the dialog by calling
18             // startResolutionForResult (),
19             // and check the result in onActivityResult ().
20             status.startResolutionForResult (
21                 OuterClass.this ,
22                 REQUEST_CHECK_SETTINGS);
23         } catch (SendIntentException e) {
24             // Ignore the error .
25         }
26         break;
```

Changing Location Settings (1/3)

```
25         case LocationSettingsStatusCodes.  
26             SETTINGS_CHANGE_UNAVAILABLE:  
27                 // Location settings are not satisfied .  
28                 // However, we have no way  
29                 // to fix the settings so we won't show the dialog .  
30                 ...  
31                 break;  
32     }  
33 });
```

Testing location-based apps with the Android emulator

Click on “...” to access extended controls



Emulator extended controls

Extended controls

Location

Cellular

Battery

Phone

Directional pad

Fingerprint

Virtual sensors

Settings

Help

GPS data point

Coordinate system: Decimal

Longitude: 56

Latitude: -3

Altitude (meters): 0.0

SEND

Currently reported location

Longitude: -3.1872
Latitude: 55.9440
Altitude: 0.0

GPS data playback

| Delay (sec) | Latitude | Longitude | Elevation | Name | Description |
|-------------|----------|-----------|-----------|---------|-------------|
| 0 | 55.9445 | -3.19216 | 0 | Point 1 | Z |
| 2 | 55.9431 | -3.18829 | 0 | Point 2 | Y |
| 2 | 55.9444 | -3.18603 | 0 | Point 3 | T |
| 2 | 55.9434 | -3.19205 | 0 | Point 4 | H |
| 2 | 55.9439 | -3.19122 | 0 | Point 5 | G |
| 2 | 55.9458 | -3.18794 | 0 | Point 6 | U |
| 2 | 55.9458 | -3.18776 | 0 | Point 7 | T |

Speed 1X

LOAD GPX/KML

Latitude: 55.9440
Altitude: 0.0

Altitude (meters)

0.0

SEND

GPS data playback

| Delay (sec) | Latitude | Longitude | Elevation | Name | Description |
|-------------|----------|-----------|-----------|---------|-------------|
| 0 | 55.9445 | -3.19216 | 0 | Point 1 | Z |
| 2 | 55.9431 | -3.18829 | 0 | Point 2 | Y |
| 2 | 55.9444 | -3.18603 | 0 | Point 3 | T |
| 2 | 55.9434 | -3.19205 | 0 | Point 4 | H |
| 2 | 55.9433 | -3.19122 | 0 | Point 5 | G |
| 2 | 55.9458 | -3.18784 | 0 | Point 6 | U |
| 2 | 55.9458 | -3.18776 | 0 | Point 7 | T |

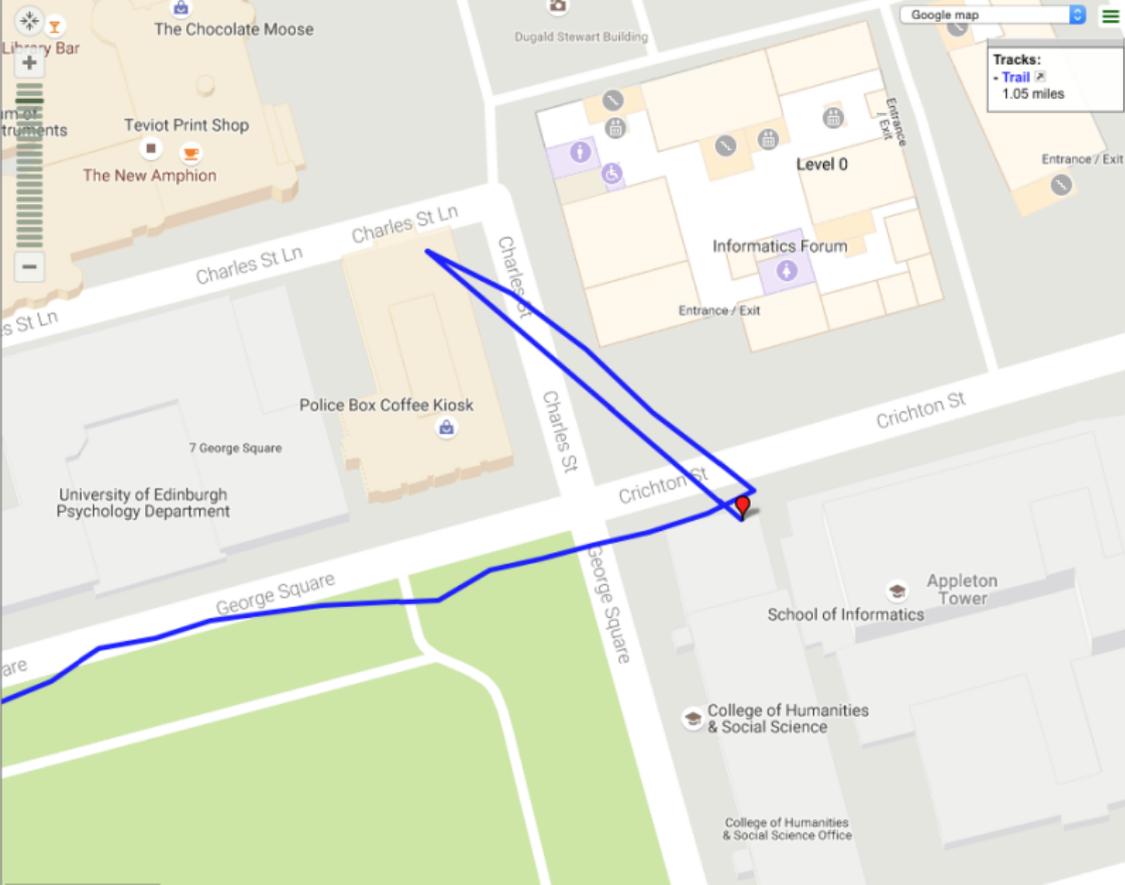


Speed 1X

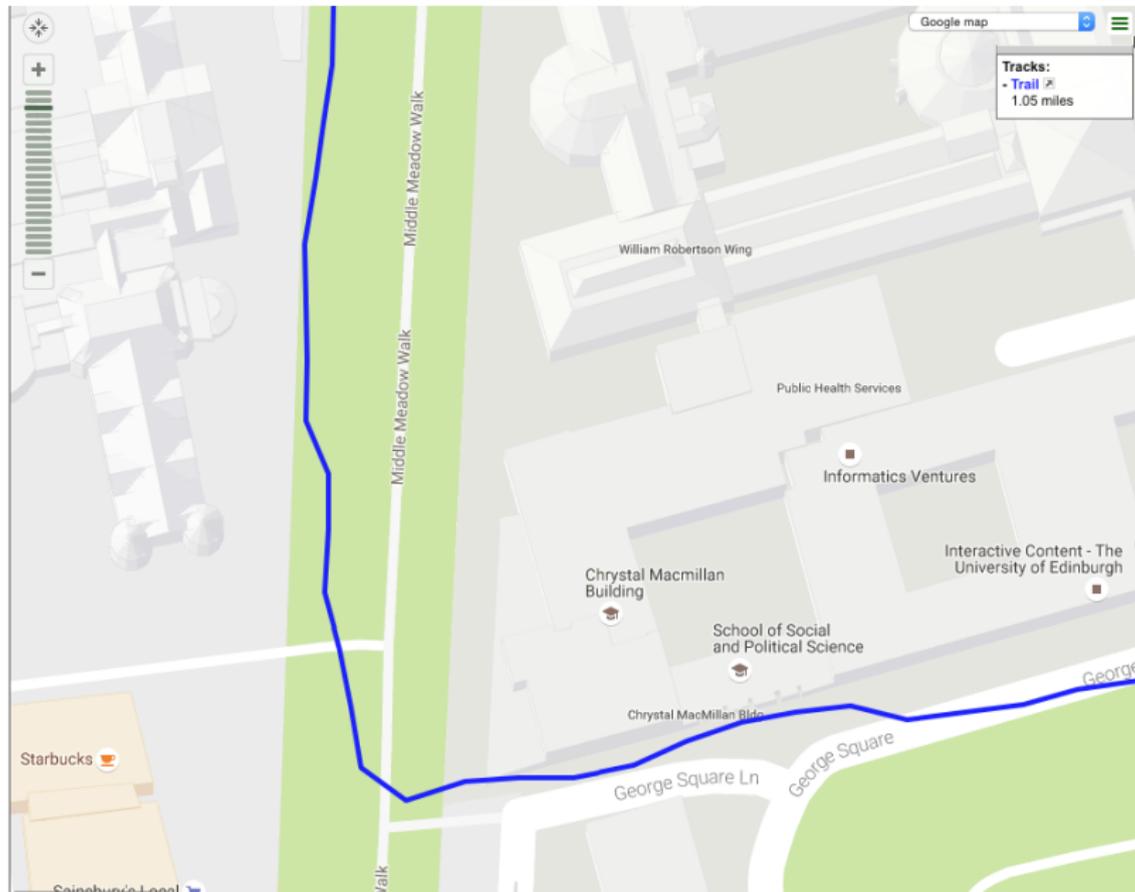


LOAD GPX/KML

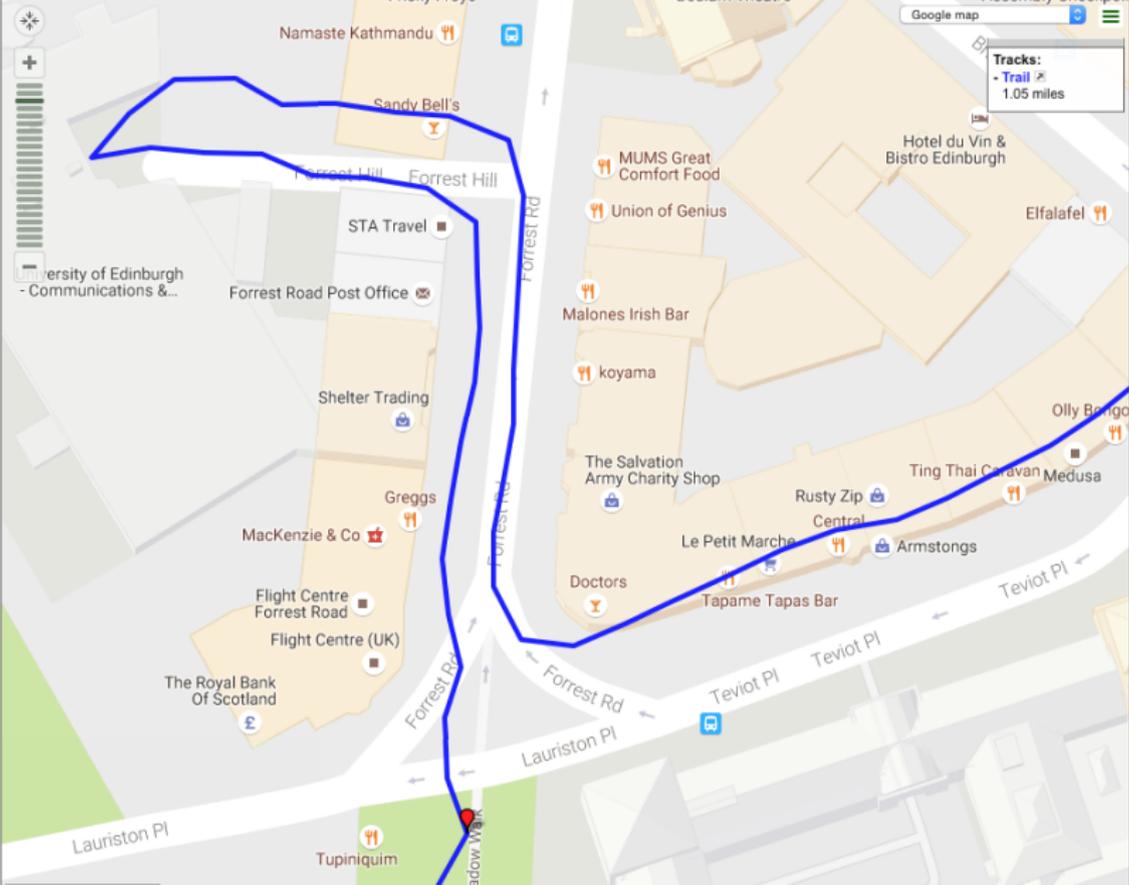
Initial GPS jitter



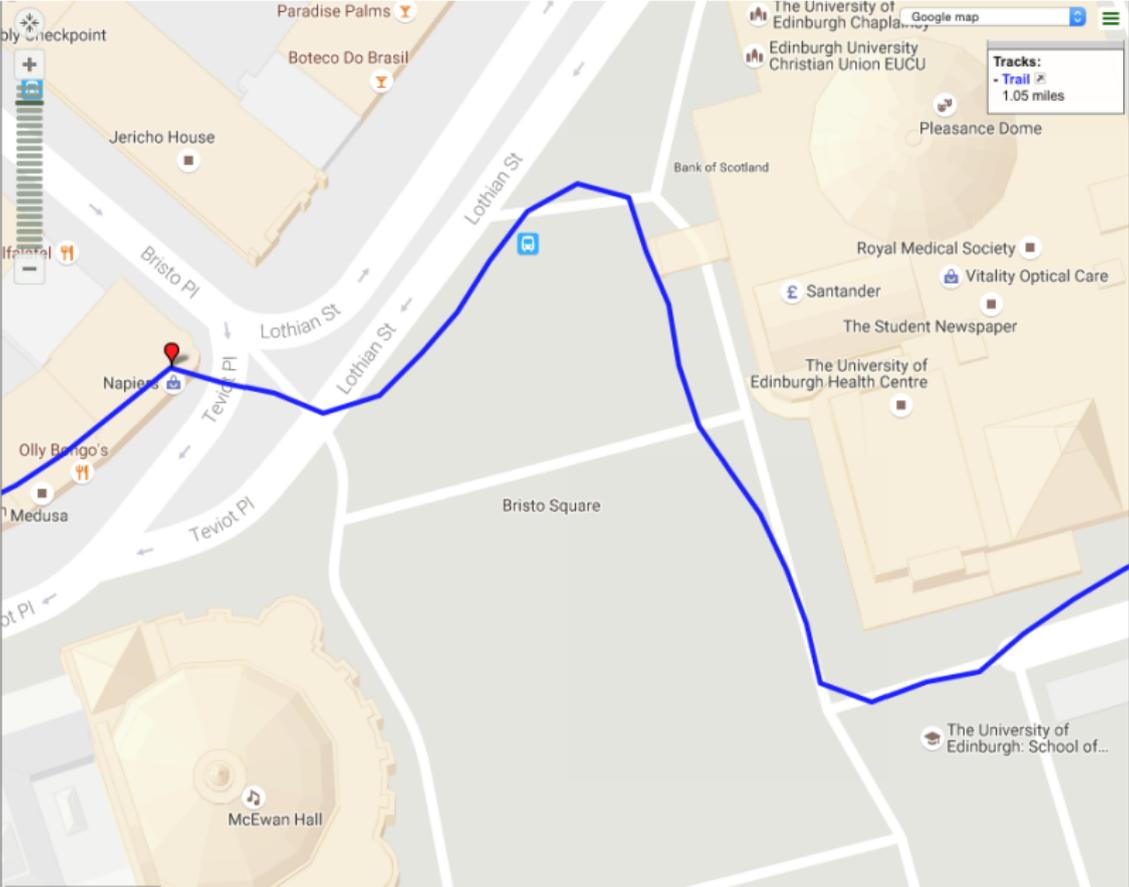
Some measurement errors seen



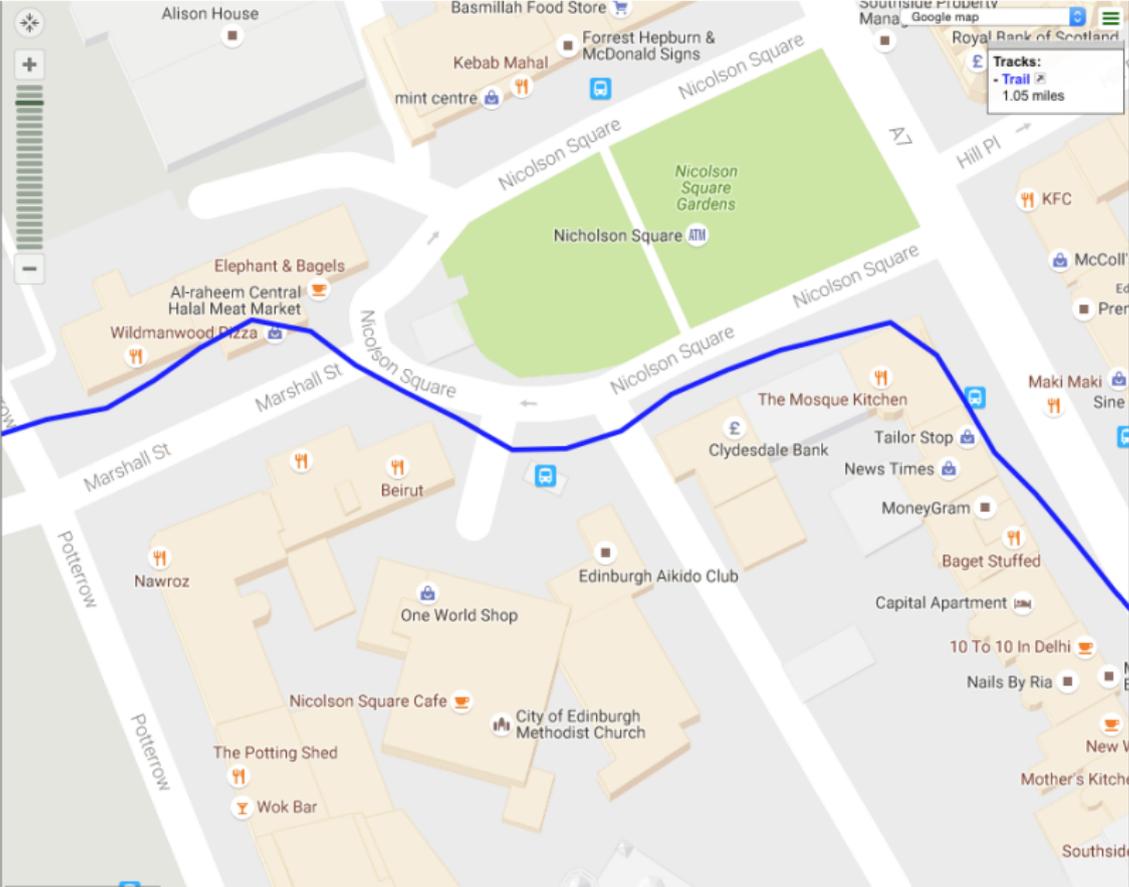
GPS measurement errors increase near tall buildings



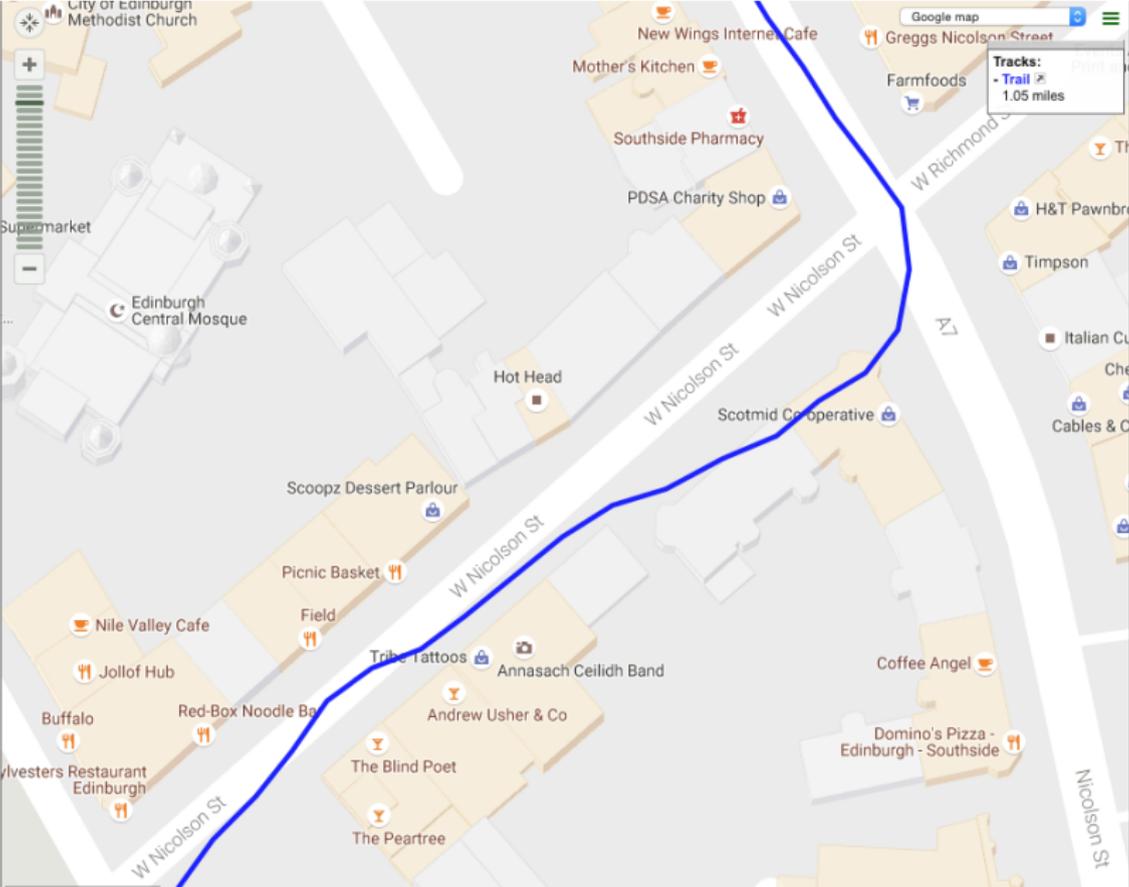
Trail continues



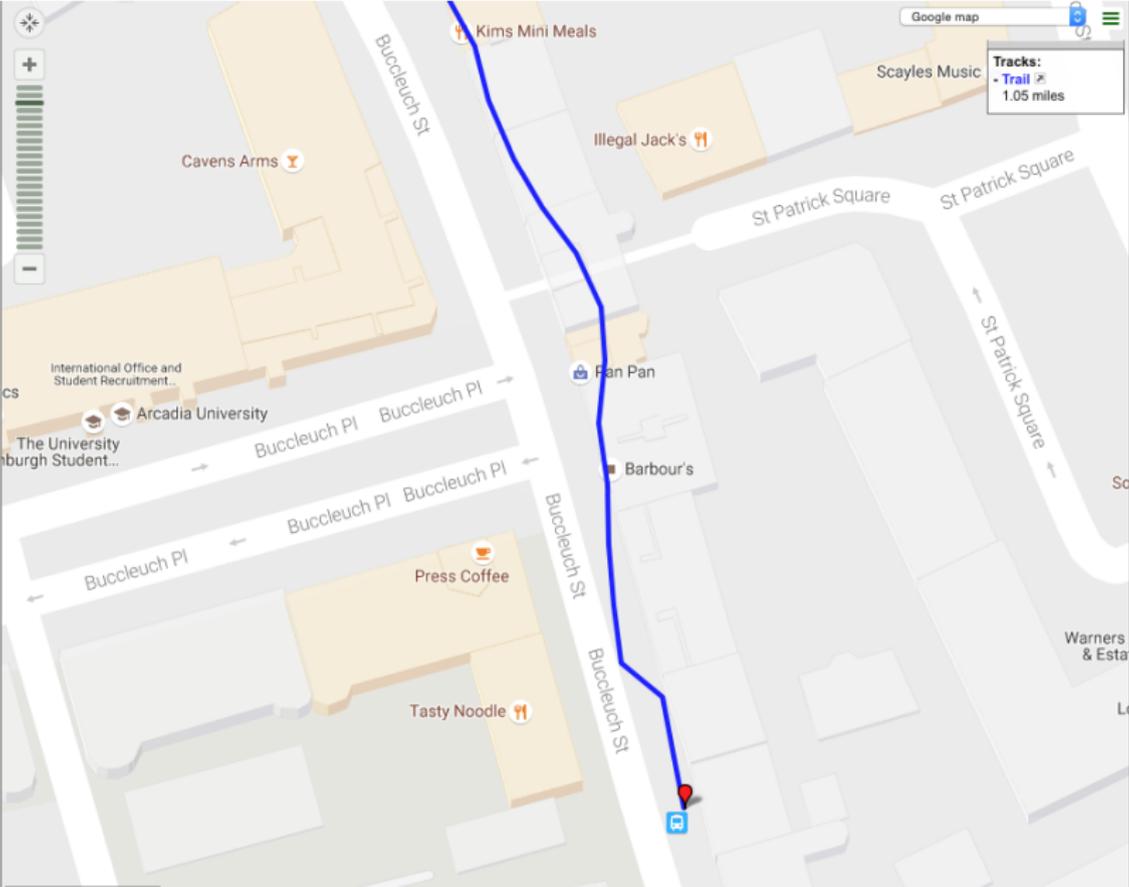
Trail continues



Trail continues



Trail ends



Concluding remarks

- ▶ This GPS trace was obtained under near-ideal weather conditions (clear sky, no cloud cover) and still contains a number of measurement errors.
- ▶ There is nothing that we can do to fix these errors, we simply take the position as reported as being the location of the user.
- ▶ The Android emulator allows us to load and replay GPS data stored in KML format. This is a useful feature for testing.

Links

- ▶ <https://developer.android.com/training/> — Android developer tutorials
- ▶ <https://github.com/googlesamples/> — Google code samples, many of which are Android-based
- ▶ <https://developer.android.com/training/location/> — Android developer tutorial on location services
- ▶ <https://github.com/googlesamples/android-play-location> — Google code sample using location services in Android