Programming
Handheld Systems
Location & Maps
Today's Topics

Location
Location support classes
Maps
Map support classes
Location Services

Mobile applications can benefit from being location-aware
Allow applications to determine & manipulate location
Using Location Information

Find stores near the user’s current location
Direct a user from a current to a particular store
Define a geofence
Initiate action when user enters or exits the geofence
Location

Represents a position on the Earth

A Location instance consists of:
Latitude, longitude, timestamp and,
optionally, accuracy, altitude, speed, and
bearing
LocationProvider

Represents a location data source

Actual data may come from

GPS satellites

Cell phone towers

WiFi access points
Location Provider Types

Network - WiFi and cell tower

GPS - Satellite

Passive - Piggyback on the readings requested by other applications
Network Provider

Determines location based on cell tower and WiFi access points

Requires either

android.permission.ACCESS_COARSE_LOCATION

android.permission.ACCESS_FINE_LOCATION
GPS Provider

Determines location using satellites

Requires

android.permission.

ACCESS_FINE_LOCATION
Passive provider

Returns locations generated by other providers

Requires

android.permission.ACCESS_FINE_LOCATION
LocationProvider

Different LocationProviders offer different tradeoffs between cost, accuracy, availability & timeliness.
Provider Tradeoffs

GPS - expensive, accurate, slower, available outdoors

Network - cheaper, less accurate, faster, availability varies

Passive - cheapest, fastest, not always available
**LocationManager**

**System service for accessing location data**

`getSystemService(Context.LOCATION_SERVICE)`
LocationManager

Determine the last known user location
Register for location updates
Register to receive Intents when the device nears or moves away from a given geographic area
LOCATIONLISTENER

Defines callback methods that are called when Location or LocationProvider status changes
LocationListener

void onLocationChanged(Location location)

void onProviderDisabled(String provider)

void onProviderEnabled(String provider)

void onStatusChanged(String provider, int status, Bundle extras)
**Obtaining Location**

**Start listening for updates from location providers**

**Maintain a "current best estimate" of location**

**When estimate is “good enough”, stop listening for location updates**

**Use best location estimate**
Determining Best Location

Several factors to consider

Measurement time

Accuracy

Power needs
LocationGetLocation

Application acquires and displays the last known locations from all providers.
If necessary, acquires and displays new readings from all providers.
Battery Saving Tips

Always check last known measurement
Return updates as infrequently as possible. Limit measurement time
Use the least accurate measurement necessary
Turn off updates in onPause()
Maps

A visual representation of area
Android provides Mapping support through the Google Maps Android v2 API
MAP TYPES

NORMAL: TRADITIONAL ROAD MAP
SATellite - AERIAL PHOTOGRAPH
HYBRID - SATELLITE + ROAD MAP
TERRAIN - TOPOGRAPHIC DETAILS
Customizing the Map

Change the camera position
Add Markers & ground overlays
Respond to gestures
Indicate the user’s current Location
Some Map Classes

GoogleMap
MapFragment
Camera
Marker
Setting up a Maps Application

Set up the Google Play services SDK
Obtain an API key
Specify settings in Application Manifest
Add map to project

See: https://developers.google.com/maps/documentation/android/start
Map Permissions

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
Map Permissions

<uses-permission android:name=
   "android.permission.WRITE_EXTERNAL_STORAGE"/>

<uses-permission android:name=
   "com.google.android.providers.
    gsf.permission.READ_GSERVICES"/>
MAP PERMISSIONS

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
MapEarthQuakeMap

This application acquires earthquake data from a server.
Then it displays the data on a map, using clickable markers.
Next Time

Data Management