

Maps

Joe Rogers
Android 310

Why use maps

- Helps user visualize location data with respect to their position.
- Provides ability to convey information faster than a list of locations in a list view.
- Even though data is provided by Google, the map itself is highly customizable.

History of maps

- Originally an “add-on” as part of the core OS.
 - Only updated when OS patched, bug fixes slow
 - Supported only Activity/View
 - API never updated and very restrictive. Essentially couldn't take advantage of new features Google Maps app.
 - Unable to take advantage of OpenGL advances

History of maps

- First version of Play Services added support for Location and Maps V2.
 - Added fragment support. Until this was done, hard to intermix maps and fragments without making a lot of hacks that were fragile.
 - New features added and on par with Google Maps App, although not a replacement.
 - Improvements, bug fixes released with each new version of Play Services.

Drawing

- Markers

- The “pin” and probably most common item to draw on an app.
- Can be modified by providing a different hue, or custom drawable.
- Info window that appears over marker when clicked is also customizable. Default is two text fields.

Drawing cont.

- Polyline
 - Create a line on the map following a series of LatLng points. Useful if plotting a custom route or shape on map.
- Polygon
 - Like a Polyline, but designed to be “closed” either automatically (or providing closed line.) However, the shape will be filled with the specified color.

Drawing cont.

- Circles
 - Draw a circle on map using LatLng and radius.
- All shapes take parameters similar to Paint
 - Stroke size, color
 - z-level (which shape should be drawn on top)
 - Geodesic indicating if lines between points in PolyLine or Polygon draw straight (default) or follow curve of earth.

Drawing Cont.

- Ground overlay
 - Place an image on the map in a specific location.
 - Image should be sized as a power of 2, or image will be “resized” to meet requirement. 512x1024
 - Image will be placed on map centered over the location with optional dimensions in “meters”
 - Advanced feature
 - Best example I can think of is “X” marks spot for treasure...

Drawing cont.

- Tile overlay
 - Used to place images over a large area on the map or extensive areas.
 - Envision plotting color over the map to represent data of some sort.
 - Highly advanced as the code is augmenting each tile with data.

Map Utilities

- Additional library providing additional support on top of the maps API
 - Cluster Map - combines markers into groups when density is too high.
 - HeatMap - Uses custom tile provider to “color” map with gradient to indicate density of data. May also be weighted if info at specific location is more intense.
 - Utilities to compute distance, area, etc.

Setup

Setup basic steps

- On Developer Console add Google Maps for Android v2 (no quota for maps)
- Add api key to manifest
 - Need api key. Because places and maps are both map related the same key may be used for both.
- Add OpenGL restriction to manifest.
 - Maps uses OpenGL version 2. This is to exclude devices not yet supporting it. (Very old devices)

Permissions

- Required
 - INTERNET
 - ACCESS_NETWORK_STATE
 - WRITE_EXTERNAL_STORAGE (cache map tiles)
- Optional
 - ACCESS_COARSE_LOCATION
 - ACCESS_FINE_LOCATION
 - Only needed if showing user's position on map.

Adding map to app

- MapFragment or SupportMapFragment
 - Easiest and best way to add map to app.
 - May derive to provide your logic (best), or optionally interact via owning activity/fragment.
- MapView
 - Able to just add map via layout.
 - Drawback is you must pass events for onCreate(), onPause(), onResume(), onDestroy(), onSaveInstanceState, and onLowMemory() to view.

References

References

- Maps V2
- Additional info on Map Utility Library