



Lezione 20



Programmazione Android



- Servizi di localizzazione
 - GPS & co.
 - Geocoding
 - Mappe



MapView



Ripasso: MapView



- Abbiamo già incontrato la MapView (brevemente) quando abbiamo parlato dei widget grafici
- **MapView** – il widget che rappresenta una mappa
- Tuttavia, l'interfaccia di questo particolare widget con il mondo è complicata
 - Interazione touch complessa
 - Dialogo con i server di Google
 - Caching
 - Visualizzazione
- Per questo motivo, spesso si preferisce *incapsularla* in componenti più completi



Maps API v1

- Nelle versioni di Android <3.0, viene fornita una Activity ad hoc
- **MapActivity** – la classe che si **deve** estendere se l'activity contiene (solo) una MapView
 - È ovviamente una sottoclasse di Activity
 - Gestisce tutto il ciclo di vita peculiare delle MapView
 - Cura il dialogo con i server di Google
- Tutto asincrono, ma largamente invisibile!



Le API key

- Varie API via web (di Google come di altri produttori) richiedono una **chiave unica dello sviluppatore o dell'applicazione**
- In particolare, per accedere alle API di Google Maps su Android, occorre registrarsi all'indirizzo
 - <http://code.google.com/android/maps-api-signup.html>
- Il processo richiede un hash MD5 della vostra chiave sviluppatore
 - Quella usata per firmare gli .apk



Usare la libreria

- La libreria che implementa MapView (e colleghi) non è parte dell'Android “standard”
 - Ergo, deve essere esplicitamente aggiunta al progetto
 - E menzionata nel manifest:

```
<uses-library  
    android:name="com.google.android.maps"  
>
```

- Servono anche i permessi di accesso alla rete

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

I GeoPoint

- È possibile controllare la visualizzazione di una MapView fornendo come parametri dei GeoPoint
 - Forma più precisa di coordinate geografiche
 - Due interi espressi in microgradi (1 milionesimo di grado)
- Conversione da lat e long a GeoPoint

```
Double lat = lat*1E6;  
Double long = long*1E6;  
GeoPoint gp =  
    new GeoPoint(lat.intValue(), long.intValue());
```
- Conversione inversa GeoPoint → coordinate

Controllo della MapView



- La classe MapView offre diversi metodi per controllare cosa viene visualizzato
- Tutti passano per un MapController associato

```
MapView mv = findViewById(...);  
MapController mc = mv.getController();
```

- Fra i tanti metodi:

```
mc.setCenter(gp); ← imposta il gp al centro  
mc.animateTo(gp); ← idem, con animazione
```



Altre caratteristiche

- La MapView fa molto di più
 - In particolare, permette di aggiungere vostri *overlay* alla mappa visualizzata
 - Percorsi, POI, ...
 - Sovraimpressioni arbitrarie
 - es., mappa meteo!
- **Non** le vedremo in dettaglio nel corso
 - Ma siete invitati a consultare la documentazione!



Maps API v2

- Il 3 Dicembre 2012, le “vecchie” API di GoogleMaps sono state **deprecate**
- È stato possibile richiedere API key per le API v1 fino al 3 Marzo 2013 – poi...
 - Le “vecchie” applicazioni continueranno a funzionare
 - Le “vecchie” chiavi saranno ancora valide
 - Nessun aggiornamento futuro alle “vecchie” API
- Le “nuove” API e key sono state rese disponibili dal Gennaio 2013, e sono ormai quelle “standard”



Maps API v2



The screenshot shows the Google Developers Console interface. The browser tabs at the top include 'Inbox - vincenzo.ger', 'Prospetto', 'Google Calendar', 'Corsi - Dipartimento', 'Presenze I term - Fo', 'Ingress Intel Map', 'Vincenzo Gervasi Ho', and 'API Library - API Pro'. The URL is <https://console.developers.google.com/project/7836250706/apiui/apis/library?authuser=0>. The page title is 'Google Developers Console' and the current view is 'API Project'. On the right, there is a 'Sign up for a free trial.' link and a user profile icon.

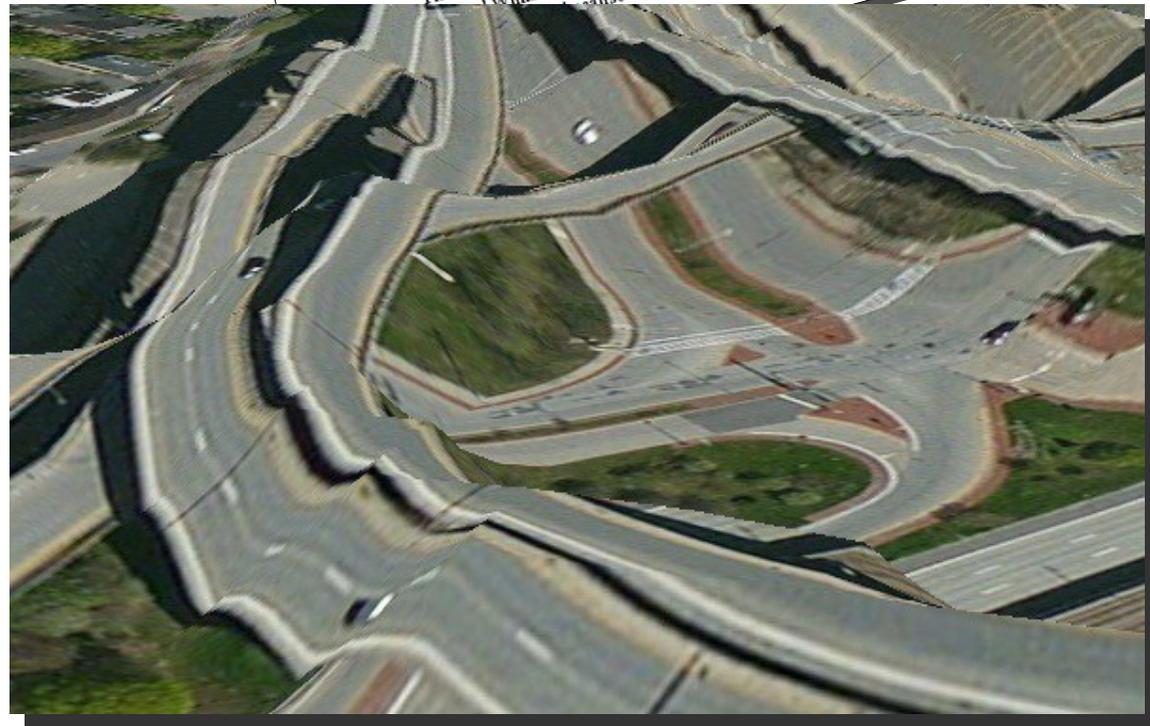
The main content area is titled 'API Library' and shows 'Enabled APIs (0)'. There is a search bar labeled 'Search all 100+ APIs'. Below this, there are several categories of APIs:

- Popular APIs**
 - Google Cloud APIs**
 - Compute Engine API
 - BigQuery API
 - Cloud Storage API
 - Cloud Datastore API
 - Cloud Deployment Manager API
 - Cloud DNS API
 - More
 - Mobile APIs**
 - Cloud Messaging for Android
 - Google Play Game Services
 - Google Play Developer API
 - Google Places API for Android
 - Advertising APIs**
 - AdSense Management API
 - DCM/DFA Reporting And Trafficking API
 - Ad Exchange Seller API
 - Ad Exchange Buyer API
 - DoubleClick Search API
 - Analytics API
 - Google Maps APIs** (highlighted with a red box and a red arrow)
 - Google Maps Android API
 - Google Maps SDK for iOS
 - Google Maps JavaScript API
 - Google Maps Embed API
 - Google Places API for Android
 - Geocoding API
 - More
 - Social APIs**
 - Google+ API
 - Blogger API
 - Google+ Pages API
 - Google+ Domains API
 - Other popular APIs**
 - Translate API
 - Custom Search API
 - URL Shortener API
 - PageSpeed Insights API
 - Fusion Tables API
 - Web Fonts Developer API
 - Google Apps APIs**
 - Drive API
 - Drive SDK
 - Calendar API
 - Gmail API
 - Google Apps Marketplace SDK
 - Admin SDK
 - More
 - YouTube APIs**
 - YouTube Data API
 - YouTube Analytics API

Maps API v2



- Alcune caratteristiche delle nuove mappe:
 - Vettoriali → minor traffico dati, migliore scalabilità
 - 3D → più figo!
 - MapFragment (non MapActivity) → maggiore flessibilità
 - Non sono le Apple Maps...





Maps API v2



- Le nuove API fanno parte dei *Google Play Services*
 - Lo stesso pacchetto che fornisce anche
 - Servizi di OAuth
 - Accesso a G+
 - Auto-aggiornamento
 - Supporto in retrocompatibilità con Android 2.2-3.0
- Purtroppo, le API sono cambiate rispetto a v1
 - Rimossa MapActivity
 - Era comunque molto limitata
 - es.: max 1 mappa per app
 - Aggiunta MapFragment
 - Se ne possono usare tante insieme
 - Riutilizzabile in vari layout

MapFragment

- La classe **MapFragment** rappresenta l'interfaccia utente “standard” per le mappe
 - Altamente configurabile: classe **GoogleMapOptions**
 - Passata a `newInstance()` di **MapFragment**
 - o al costruttore di **MapView**
 - Fra le altre opzioni:
 - Posizione iniziale della camera
 - Abilitazione o meno delle gesture: zoom, rotate, scroll, tilt
 - Abilitazione o meno della UI: pulsanti per zoom ecc.
- La mappa in sé è invece rappresentata da un oggetto **GoogleMap** (associato al fragment)

Esempio di UI

trulia 113 found | For Sale **Save** 🔍 ⌵ 👤 ☰

- 19 photos 2 br / 2 ba / 1,307 sqft Condo
- \$849,000**
501 Beale St #5G
2 br
Condo
- 17 photos
- \$949,000**
733 Front St #606
1 br / 1 ba
Condo
- 20 photos
- \$599,000**
75 Folsom St #805
1 br / 1 ba / 804 sqft
Condo
- 9 photos
- \$1,259,900**
301 Main St #16F
2 br / 2 ba / 1,307 sqft
Condo
- 11 photos
- \$1,459,000** ⬆️\$140K
333 Bush St #3805
2 br / 2 ba / 1,785 sqft
Condo
- 22 photos
Open House

Google Handbags Dress Accessories Men's Furnishings

©2012 Google - Map data ©2012 Google, Sanborn

L'oggetto GoogleMap

- Per ottenere un GoogleMap, si può chiamare il metodo `getMap()` del `MapFragment`
 - Il metodo può fallire e restituire null per molte ragioni
 - Per esempio, il Fragment non è ancora stato inizializzato
 - Oppure, non sono installati i Google Play Services
 - Oppure, la vostra chiave sviluppatore non è valida
 - Oppure, Saturno è in trigono con Urano nei Pesci e oggi è venerdì
 - In questi casi, occorre in qualche modo avvisare l'utente
 - Si può anche ritentare l'operazione più avanti



L'oggetto GoogleMap Cautele



- GoogleMap è un oggetto molto *delicato*
 - Mantiene moltissimi dati
 - Quindi bisogna evitare di tenere riferimenti inutili
 - Anche a oggetti derivati, come i Marker
 - Altrimenti, si blocca il Garbage Collector
 - Effettua numerose operazioni via rete
 - Comportamento asincrono sulla UI
 - Può essere acceduto **solo** dal thread UI
 - `runOnUiThread()`, `post()`, `postDelayed()`
 - Può essere necessario usare **synchronized**



Funzioni offerte da GoogleMap



- In cambio, GoogleMap fornisce metodi per
 - Sovrapporre vostri contenuti alla mappa
 - `addGroundOverlay()`, `addMarker()`, `addPolygon()`, ...
 - `setInfoWindowAdapter()`
 - Gestire la visuale e l'aspetto
 - `getCameraPosition()`, `moveCamera()`, `animateCamera()`, ...
 - `setMapType()`, `setTrafficEnabled()`, `setMyLocationEnabled()`, `setIndoorEnabled()`, ...
 - Mappare fra coordinate geografiche e pixel su schermo
 - `getProjection()`



Funzioni offerte da GoogleMap



- In cambio, GoogleMap fornisce metodi per
 - Registrare un certo numero di listener
 - `setOnCameraChangeListener()` → spostamenti camera
 - `setOnInfoWindowClickListener()` → click sui pop-up
 - `setOnMapClickListener()` → click sulla mappa (punti scoperti)
 - `setOnMapLongClickListener()` → long-click sulla mappa
 - `setOnMarkerClickListener()` → click su un marker
 - `setOnMarkerDragListener()` → drag di un marker
 - Gli spostamenti di camera consentono anche di posizionare la (parte visibile della) mappa

Posizionamento camera



- Nelle Maps v1, si impostava il centro mappa (lat,long), lo zoom, e il gioco era fatto
- Nella Maps v2, si costruiscono istruzioni per lo spostamento della camera tramite la classe **CameraUpdate**
 - Quasi una sceneggiatura!
 - I CameraUpdate si applicano poi con
 - animateCamera() → animazioni gestite autonomamente
 - moveCamera() → spostamento immediato



Posizionamento camera



- I `CameraUpdate`, a loro volta, non vengono creati direttamente, ma prodotti da una classe factory
 - Ovviamente, **`CameraUpdateFactory`**
- Questa fornisce metodi per
 - Creare update che spostano la camera
 - `newCameraPosition()`, `newLatLng()`, ...
 - Creare update che cambiano l'inquadratura
 - `zoomBy()`, `zoomIn()`, `zoomOut()`, `zoomTo()`
 - `scrollBy()`



Posizionamento camera

Esempio



- Vogliamo animare uno spostamento della camera che porti al centro della mappa le coordinate *lat* e *lng*, mantenendo lo zoom corrente

```
MapFragment mfrag = ... ;  
GoogleMap gmap = mfrag.getMap() ;  
if (gmap!=null) {  
    LatLng ll = new LatLng(lat,lng) ;  
    gmap.animateCamera(CameraUpdateFactory.newLatLng(ll)) ;  
} else {  
    /* aita! */  
}
```

- Se vi sembra un po' barocco...
 - ... avete ragione! Però è anche potente



I marker

- Come abbiamo detto, è possibile aggiungere propri **Marker** a una mappa
- Si passa a `addMarker()` un oggetto `MarkerOptions` che specifica che marker vogliamo aggiungere
- `addMarker()` restituisce un oggetto `Marker` che possiamo poi usare per riferire il marker creato
 - Un tap sul marker avrà l'effetto di centrare la mappa sulla sua posizione



I marker



- Un `MarkerOptions` specifica
 - Informazioni testuali
 - Titolo, snippet
 - Il pop-up di default mostra questi testi, ma può essere sostituito chiamando `setInfoWindowAdapter()` sulla mappa
 - Informazioni grafiche
 - Icona, offset (u,v) dell'hot-point dentro l'icona
 - Informazioni geografiche
 - Coordinate geografiche, espresse con un `LatLng`
 - Comportamento
 - Visibile, draggabile



I marker



trulia 113 found | For Sale **Save**

19 photos 2 br / 2 ba / 1,307 sqft Condo

\$849,000
501 Beale St #5G
2 br
Condo

17 photos

\$949,000
733 Front St #606
1 br / 1 ba
Condo

20 photos

\$599,000
75 Folsom St #805
1 br / 1 ba / 804 sqft
Condo

9 photos

\$1,259,900
301 Main St #16F
2 br / 2 ba / 1,307 sqft
Condo

11 photos

\$1,459,000 \uparrow \$140K
333 Bush St #3805
2 br / 2 ba / 1,785 sqft
Condo

22 photos
Open House

Map markers: \$1.1M, \$4.3M, \$949K, \$1.9M, \$628K, \$1.5M, \$1.6M

Streets: California St, Vinton Ct, Notre Dame Des Victoires Church, Sutter St, Campion St, Post St, Stockton St, Grant Ave, Annie St, 3rd St, 2nd St, 595 Market Street, McKesson Plaza, Mechanics' Institute, Men's Furnishings, Handbags, Dress Accessories

©2012 Google - Map data ©2012 Google, Sanborn

I marker

- Un `InfoWindowAdapter` deve implementare due metodi
 - `View getInfoWindow(Marker m)`
 - Dato un marker, restituisce una `View` per l'**intero** pop-up
 - Può restituire `null`, nel qual caso viene chiamato invece...
 - `View getInfoContents(Marker m)`
 - Dato un marker, restituisce una `View` per il **contenuto** del pop-up (che viene inserito nella finestrella standard)
 - Può restituire `null`, nel qual caso viene usato il pop-up di default (titolo e snippet)



Marker Esempio



```
public class MainActivity extends Activity {  
    static final LatLng FIBO = new LatLng(43.7199949,10.4073628);  
    private GoogleMap map;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        map = ((MapFragment)getFragmentManager().findFragmentById(R.id.map)).getMap();  
  
        if (map!=null) {  
            Marker mfibo = map.addMarker(new MarkerOptions()  
                .position(FIBO)  
                .title("Polo Fibonacci")  
                .snippet("Che figo studiare SAM qui!")  
                .icon(BitmapDescriptorFactory.fromResource(R.drawable.tocco))  
                .alpha(0.8)  
            );  
        }  
    }  
}
```

Copy&Paste dalle URL di
Gmaps web

com.google.android.gms.maps.model.BitmapDescriptorFactory