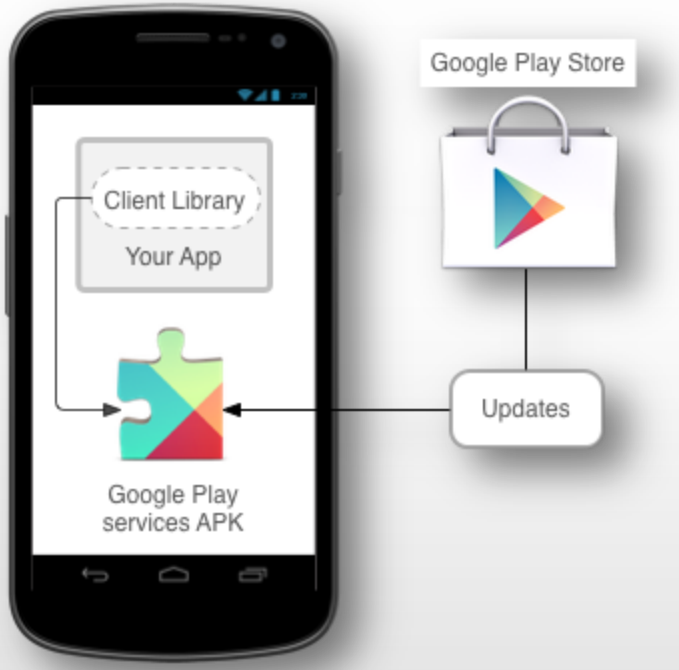
ANDROID MAPPING & LOCATION

Andrew Thompson & Scott Samson

MAPPING IN ANDROID

- Google Maps API v2
  - Not just for Android!
    - Web Services
    - iOS
  - Map Images, Google Earth
  - Google Places API
  - Google Directions API

- Google Play Services
  - Need to add to Android SDK
  - Add library to your project
    - com.google.android.gms



LOCATION BASED SERVICES (LBS)

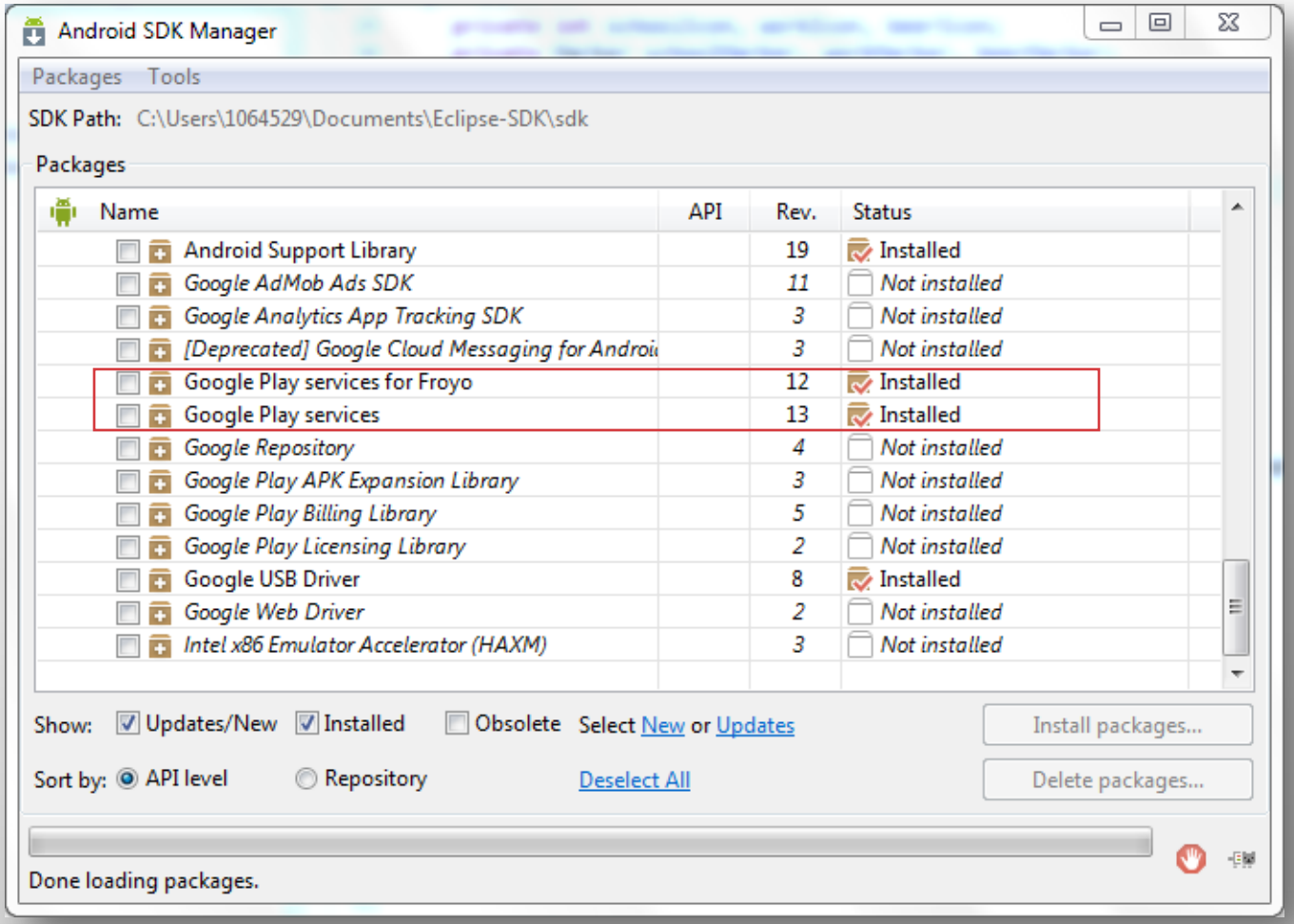
- Types of Location Based Services
  - Cell ID
  - Triangulation
  - GPS
    - Downsides: Increased Cost, Reduced Battery life
- Challenges
  - Many location sources
  - User movement
  - Varying accuracy

LOCATION APIS

- Fused location provider
  - Simple API
  - Immediately Available
  - Power-efficient
  - Versatile
- Activity Recognition
  - Optimized for battery life
  - Enhances other services with context
  - Features for Advanced Applications
- Geofencing APIs
  - Simple but powerful
  - Optimized for battery life

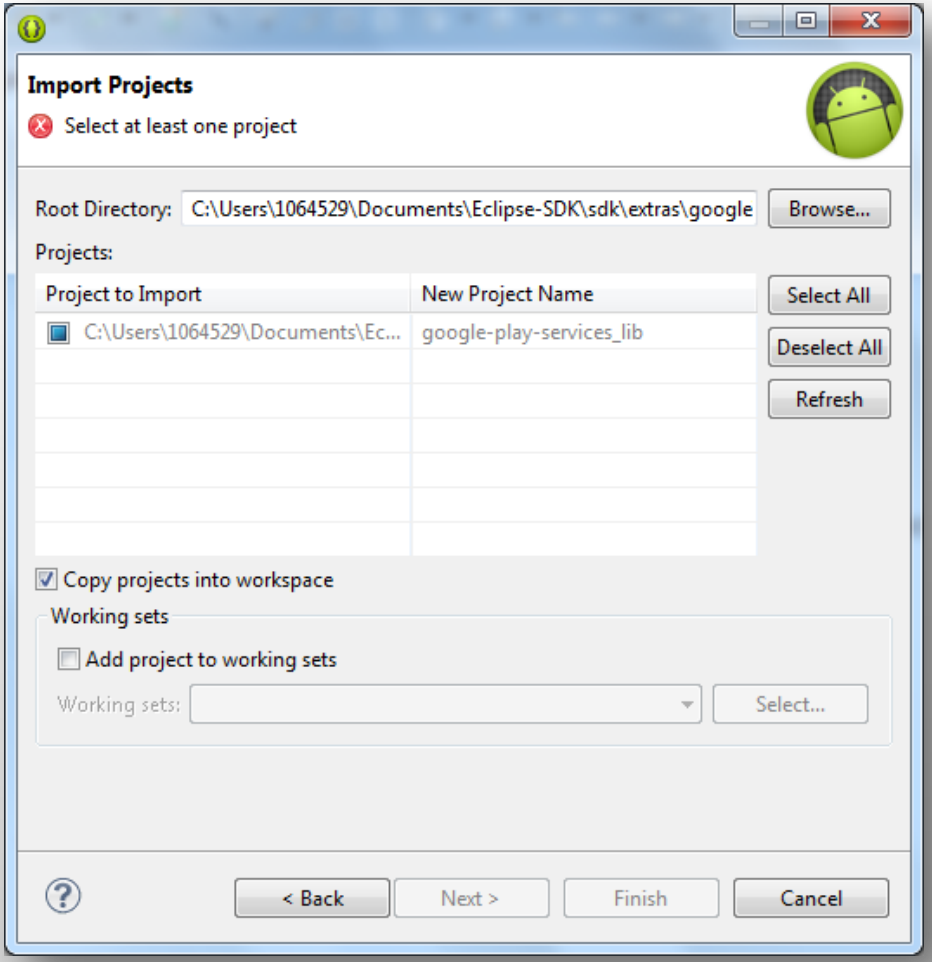
ADD GOOGLE PLAY SERVICES SDK

- Install the Google Play services SDK.



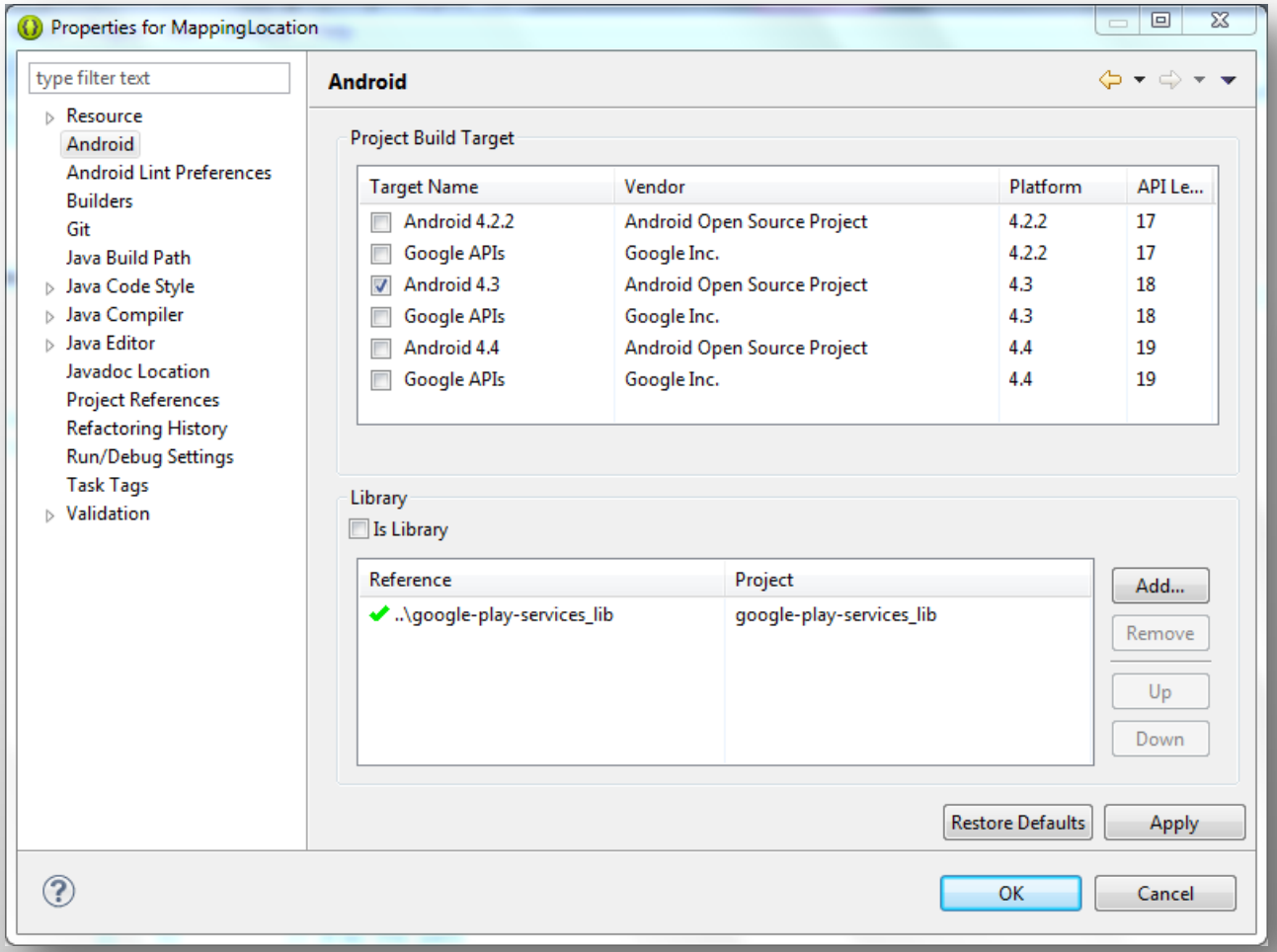
ADD GOOGLE PLAY SERVICES SDK

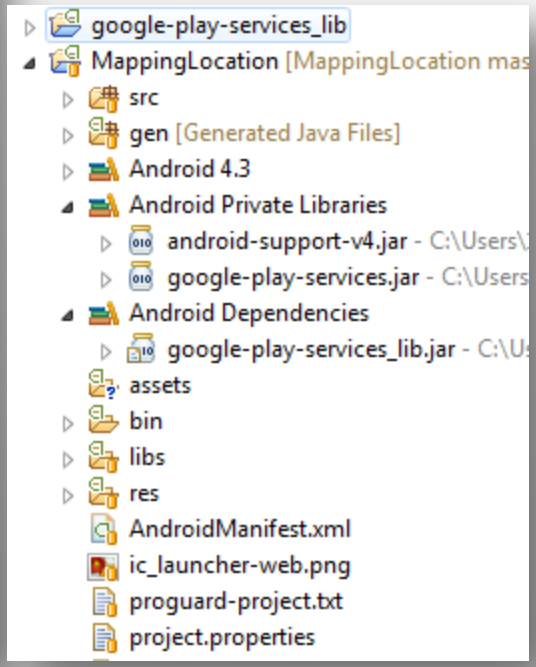
- Add Google Play services as an Android library project.



ADD GOOGLE PLAY SERVICES SDK

- Reference the Google Play services in your app's project.





MANIFEST FILE

- Permissions

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

  - **ACCESS_COARSE_LOCATION**
    - Use WiFi and/or mobile cell data to determine the device's location.
  - **ACCESS_FINE_LOCATION**
    - Use GPS to determine the device's location to within a very small area.

- Google API Key

```
<application>
    ...
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="YOUR_KEY_GOES_HERE"/>
    ...
</application>
```

MAP OPTIONS & METHODS

- Options
  - Set map type
    - Hybrid, Normal, Satellite, Terrain, None
  - Manipulate UI Elements
    - Location Button
    - Zoom Controls
    - Traffic
- User Response & Events
  - Camera and location change
  - Click and long click on map
  - Marker click or drag
- Camera
  - Animate camera
  - Move camera

```
// setup the original camera position
camOriginal = new CameraPosition.Builder()
    .target(sdsmtLocation)
    .zoom(15)
    .tilt(0)
    .bearing(0)
    .build();

// move the camera
googleMap.moveCamera(CameraUpdateFactory.newCameraPosition(camOriginal));
```

MAP LAYOUT

- Google Maps API v1 deprecated December 3, 2012
- Biggest change is switch to fragments

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <fragment xmlns:android="http://schemas.android.com/apk/res/android"
3            android:id="@+id/map"
4            android:layout_width="match_parent"
5            android:layout_height="match_parent"
6            android:name="com.google.android.gms.maps.MapFragment"/>
```

INITIALIZE THE MAP

```
30      // map and location member
31      private GoogleMap googleMap;
32      private MapFragment mapFragment;
```

```
45      @Override
46      protected void onCreate(Bundle savedInstanceState)
47      {
48          super.onCreate(savedInstanceState);
49          setContentView(R.layout.activity_google_map);
50
51          // initialize the map
52          if(googleMap == null)
53          {
54              // get the map fragment and the map object
55              mapFragment = (MapFragment) getFragmentManager().findFragmentById(R.id.map);
56              googleMap = mapFragment.getMap();
57
58              // make sure the map exists before using
59              if(googleMap != null)
60              {
61                  // do map work here
62              }
63          }
64      }
```

MAP MARKERS AND PATHS

- Marker

```
// setup a map marker and add it to the map
Marker myMarker = googleMap.addMarker(new MarkerOptions()
    .position(new LatLng(44.0748167, -103.2058167))
    .title("Location Name")
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.bluedot))
    .snippet("Short Description")
    .visible(false));
```

- Polyline

```
// define the path and add it to the map
Polyline polyline = googleMap.addPolyline(new PolylineOptions()
    .color(Color.parseColor("#755673DB"))
    .add(new LatLng(44.07626, -103.2073),
         new LatLng(44.076616, -103.20715),
         new LatLng(44.07803, -103.2104),
         new LatLng(44.079, -103.21425),
         new LatLng(44.081683, -103.230716),
         new LatLng(44.080616, -103.23103),
         new LatLng(44.080216, -103.22856)));
```

OTHER OBJECTS

- Polygon

```
// define the polygon and add it to the map
Polygon polygon = googleMap.addPolygon(new PolygonOptions()
    .add(new LatLng(44.07615744242147, -103.20732861757278),
         new LatLng(44.07583756928344, -103.2060518860817),
         new LatLng(44.075434834982005, -103.2062503695488),
         new LatLng(44.075704609458285, -103.20736885070801),
         new LatLng(44.07591464730614, -103.20743322372437))
    .strokeWidth((float) 1.0)
    .strokeColor(Color.parseColor("#BEFF0000"))
    .fillColor(Color.parseColor("#40FF0000")));
```

- Circle

```
// define the circle and add it to the map
Circle circle = googleMap.addCircle(new CircleOptions()
    .fillColor(Color.parseColor("#255673DB"))
    .strokeColor(Color.parseColor("#905673DB"))
    .strokeWidth((float) 2.0)
    .center(new LatLng(44.07626, -103.2073))
    .radius(100));
```

LOCATION WITHOUT MAPPING

- Location Manager
- Can use any provider type

```
// find the TextViews
TextView tvLatitude = (TextView)findViewById(R.id.tvLatitude);
TextView tvLongitude = (TextView)findViewById(R.id.tvLongitude);

// get handle for LocationManager
LocationManager lm = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);

// connect to the GPS location service
Location loc = lm.getLastKnownLocation("gps");

// fill in the TextViews
tvLatitude.setText(Double.toString(loc.getLatitude()));
tvLongitude.setText(Double.toString(loc.getLongitude()));
```

LOCATION PROVIDER

- Allows automatic updates
- LocationManager implements LocationListener
  - OnLocationChanged
  - OnProviderEnabled
  - OnProviderDisabled
  - OnStatusChanged
- requestLocationUpdates(provider, minTime, minDist, listener);

```
// ask the Location Manager to send us location updates
locListenD = new DispLocListener();
lm.requestLocationUpdates("gps", 30000L, 10.0f, locListenD);
```

STREETVIEW

- Google encourages use of its StreetView activity
- Use intent to launch via URI
- cbp and mz are optional
- google.streetview:cbll=lat,lng&cbp=1,yaw,,pitch,zoom&mz=mapZoom

```
String uri = "google.streetview:cbll=42.352299,-71.063979&cbp=1,0,,0,1.0&mz=12";
Intent streetView = new Intent(android.content.Intent.ACTION_VIEW, Uri.parse(uri));
startActivity(streetView);
```

SOURCES

- Maps
  - https://developers.google.com/maps/documentation/android/
  - http://mobile.tutsplus.com/tutorials/android/android-sdk-working-with-google-maps-application-setup/
  - http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/
- Location
  - http://developer.android.com/google/play-services/location.html
  - http://developer.android.com/guide/topics/location/strategies.html

Mednieks, Zigurd, Laird Dornin, G. Blake. Meike, and Musami Bakamura. "Chapter 15: Location and Mapping." Programming Android. Sebastopol, CA: O'Reilly, 2012.