CS 646 Android Mobile Application Development
Spring Semester, 2015
Doc 18 Location & Maps
April 14, 2015

Chapter in Big Nerd Ranch Text

Chapter 33

Android - Google View

Google gives Android away for free

- Except for standard Google Apps

Make Android a commodity

- Everyone has smart phone

Ensures people use Google search on phones

Android - Phone Manufactures View

Free-ish OS

Need to modify Android interface

Differentiate from other manufactures

Do not want product to be a commodity

Update Dilemma

Updating OS on existing phones

- Cost phone manufactures money
- May decrease sales of new phones in short run

So how does Google proved new services to existing Android phones?

Google Play App

Provide services through Google play app

Updating Google Play app

- Updates Google APIs on existing phones

Android Freeloaders

Android forks

- Amazon Tablets
- Some Chinese manufactures

Devices with Android app compatibility

- Blackberry
- Tinzen
- Bada
- Sailfish

Google Play Services Side Effect

Apps using the new Google Services will not run on the Android Freeloaders

Setting Up Google Play Services

Instructions

http://developer.android.com/google/play-services/setup.html

Location

Location

Uses Google Play Service

Google Tutorial

http://developer.android.com/training/location/retrieve-current.html

Permissions

ACCESS_COARSE_LOCATION

- Uses Cell Tower & WiFi
- City block accuracy

ACCESS_FINE_LOCATION

- More accurate
- Uses more power

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.google.android.gms.location.sample.basiclocationsample" >

  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
</manifest>
```

Getting Last Location

Class getting the Location data needs to implement

ConnectionCallbacks.ConnectionCallbacks

onConnected(Bundle connectionHint)

Called when connection is made

onConnectionSuspended(int cause)

Called when temporarily disconnected

ConnectionCallbacks.OnConnectionFailedListener

onConnectionFailed(ConnectionResult result)

Called when can not connect to client

Getting Last Location - Connecting to Service

```
mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
```

Getting Last Location - Getting Last Location

```
public void onConnected(Bundle connectionHint) {
        mLastLocation = LocationServices.FusedLocationApi.getLastLocation(
                mGoogleApiClient);
        if (mLastLocation != null) {
            mLatitude = String.valueOf(mLastLocation.getLatitude());
            mLongitude = String.valueOf(mLastLocation.getLongitude()));
        }
    }
```

Getting Location Updates

In addition to OnConnectionFailedListener & ConnectionCallbacks must implement

LocationListener

onLocationChanged(Location location)

Location Requests

Specify

update interval

fastest update interval app can handle

priority

PRIORITY_BALANCED_POWER_ACCURACY

~100 meters

PRIORITY_HIGH_ACCURACY

PRIORITY_LOW_POWER

~10Km

PRIORITY_NO_POWER

Only get results when other apps request location

Location Requests - Connecting to Service

```
mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
```

Location Requests - Getting the Request

```
public void onConnected(Bundle connectionHint) {
    LocationRequest accurateRequest = new LocationRequest();
    accurateRequest.setInterval(10000);
    accurateRequest.setFastestInterval(5000);
    accurateRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

    LocationServices.FusedLocationApi.requestLocationUpdates(
            mGoogleApiClient, mLocationRequest, this);
    }

public void onLocationChanged(Location location) {
        mLatitude = String.valueOf(location.getLatitude());
        mLongitude = String.valueOf(location.getLongitude());
    }
```

Maps

Google Maps v2

Uses Google Play API

Need free API Key

Installation

Hardest part of using maps

Follow the instructions

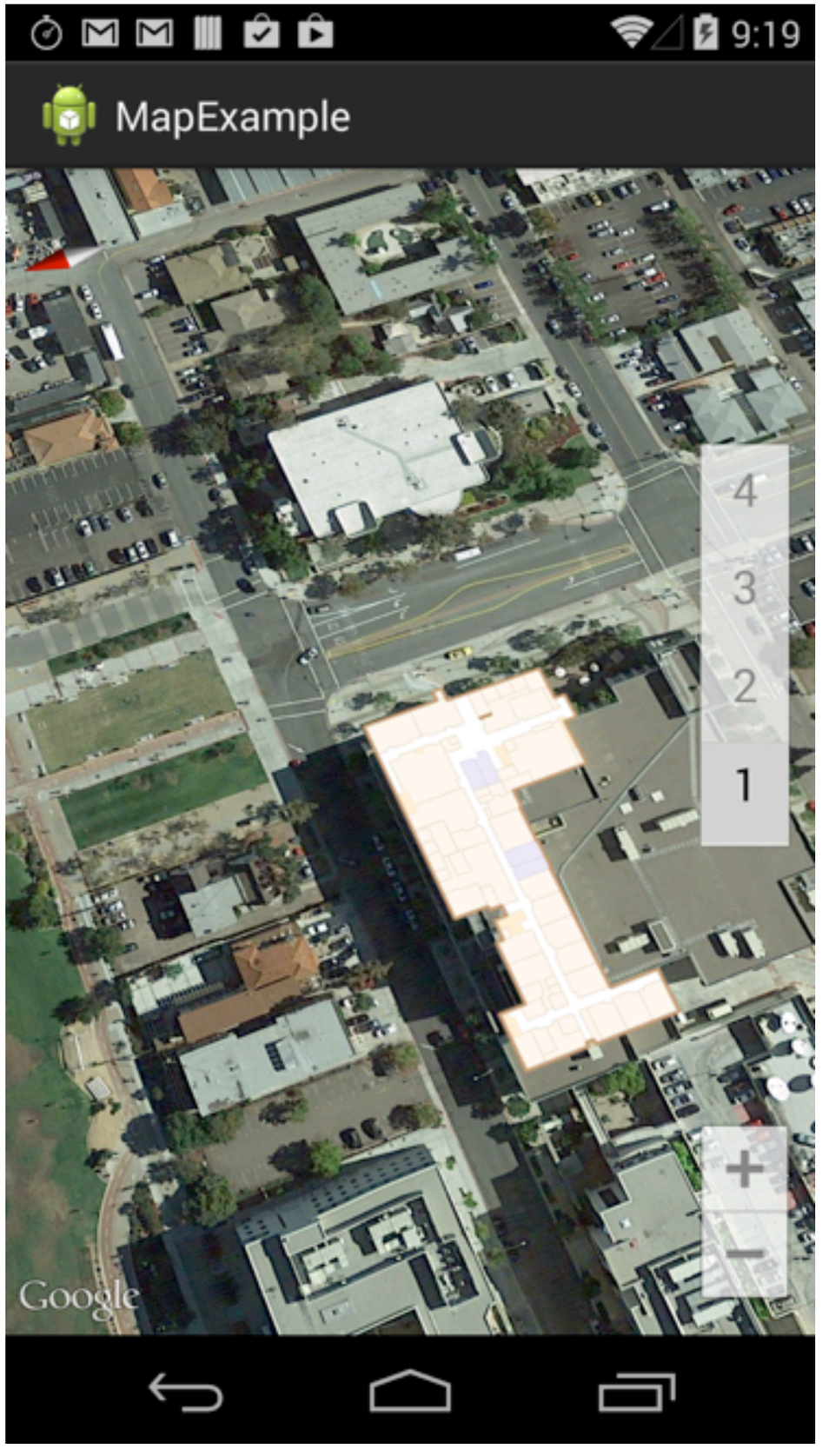
https://developers.google.com/maps/documentation/android/start

Map Example

Requires

google-play-services_lib project

Follow Google instructions



https://developers.google.com/maps/documentation/android/start

Added to Manifest File

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />

<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" />
```

Also Added to Manifest file

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="YourMapKeyGoesHereThisIsNotAValidKey" />
```

All these were given in Google Map install instructions

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    map:cameraBearing="112.5"
    map:cameraTargetLat="32.772399"
    map:cameraTargetLng="-117.072286"
    map:cameraTilt="30"
    map:cameraZoom="18"
    map:mapType="satellite"
    map:uiCompass="true"
    map:uiRotateGestures="true"
    map:uiScrollGestures="true"
    map:uiTiltGestures="true"
    map:uiZoomControls="true"
    map:uiZoomGestures="true" />
```

Activity

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

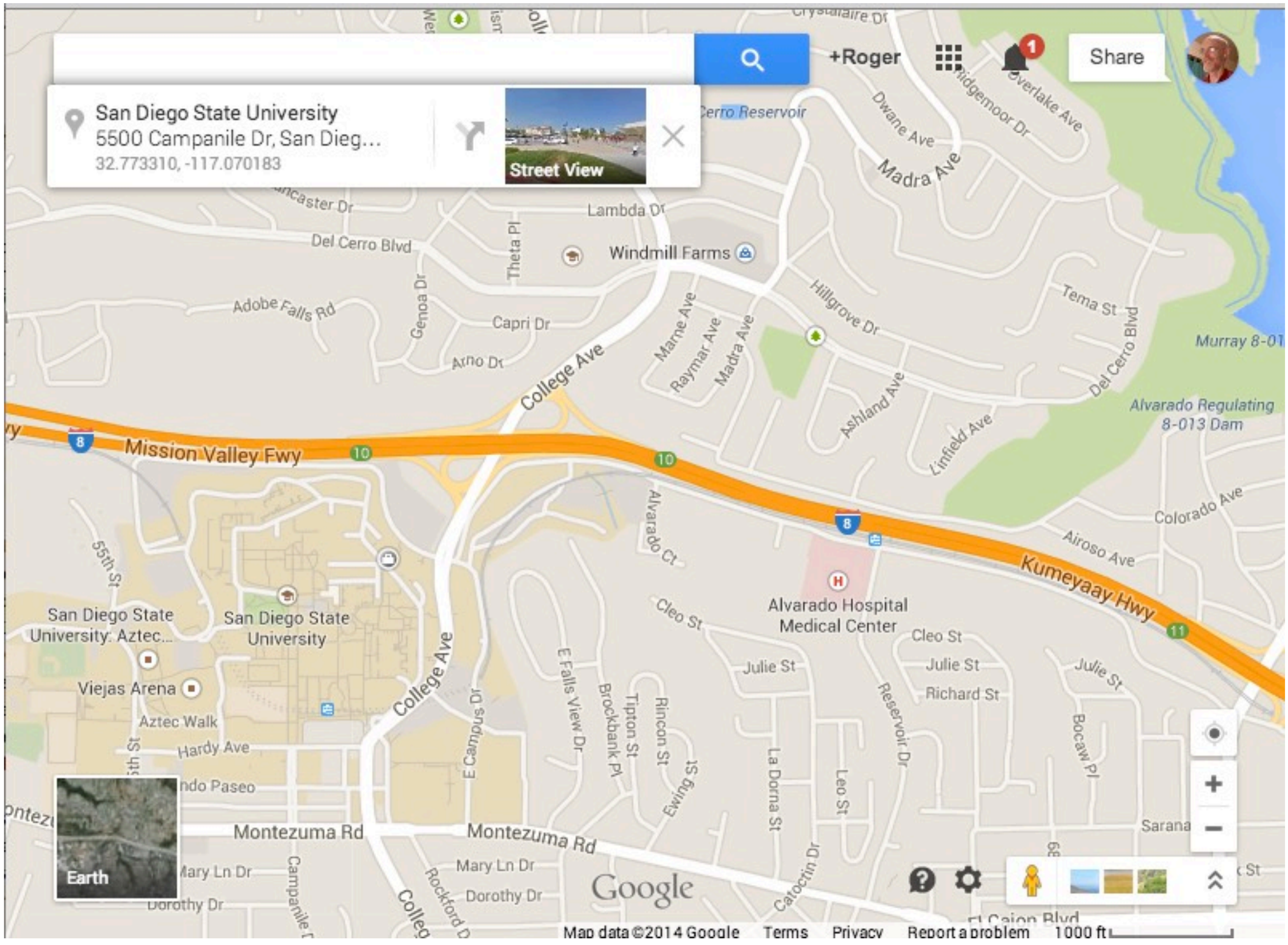
Map Attributes

map:cameraBearing="112.5"
map:cameraTargetLat="32.772399"
map:cameraTargetLng="-117.072286"

Can be set in

- Layout
- Code

Latitude, Longitude



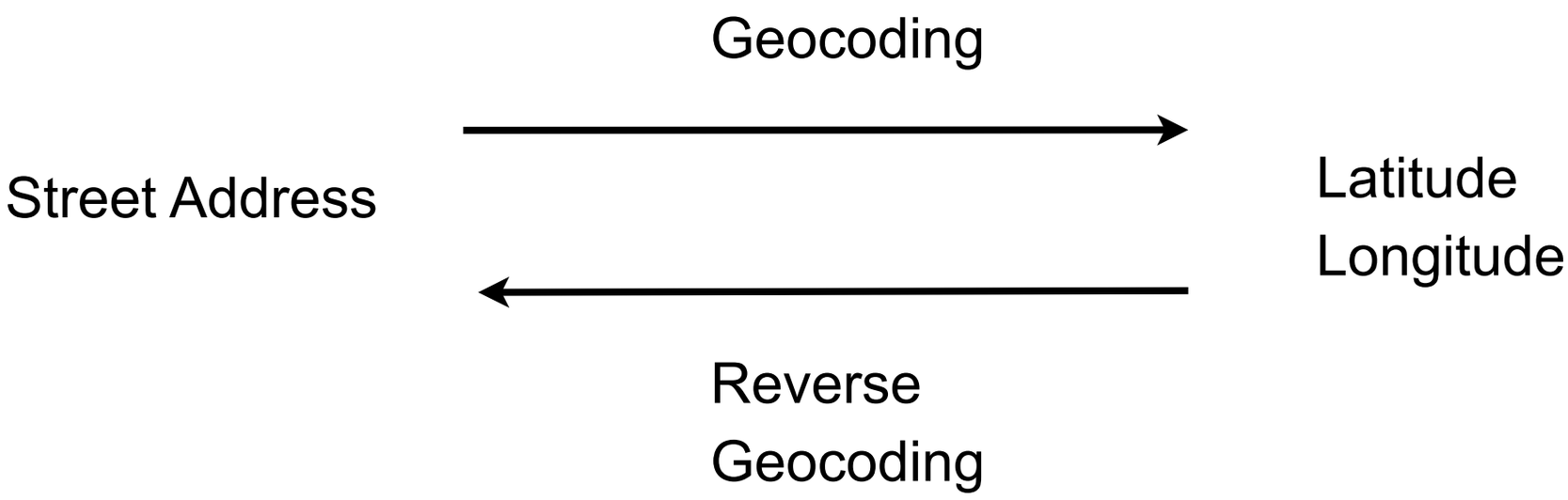
Latitude, Longitude

Normal people do not know latitude or longitude of any location

They do not want to know

Do not ask them to enter a location by latitude/longitude

Geocoding


Geocoding
Street Address
Latitude
Longitude
Reverse
Geocoding

android.location.Geocoder

Does Geocoding and Reverse Geocoding

Has 4 methods

getFromLocation(double latitude, double longitude, int maxResults)

getFromLocationName(String locationName, int maxResults, double lowerLeftLatitude, double lowerLeftLongitude, double upperRightLatitude, double upperRightLongitude)

getFromLocationName(String locationName, int maxResults)

isPresent()

getFromLocation

Returns list of possible addresses

```
Geocoder location = new Geocoder(this);
try {
    List<Address> sdsu = location.getFromLocation(32.772399, -117.072286,
5);
    for (Address street: sdsu) {
      int index = 0;
      while (street.getAddressLine(index) != null) {
          Log.i("rew", "line " + index + " = " + street.getAddressLine(index));
          index++;
      }
}
```

Output

```
line 0 = 5250 Campanile Dr
line 1 = San Diego State University
line 3 = USA

line 0 = San Diego, CA 92182
line 1 = USA

line 0 = College West
line 1 = San Diego, CA
line 2 = USA

line 0 = Mid-City
line 1 = San Diego, CA
line 2 = USA

line 0 = San Diego, CA
line 1 = USA
```

android.location.Address

Set of Strings describing a location

Contains enough fields to describe any location on the planet

Zoom

map:cameraZoom="18"

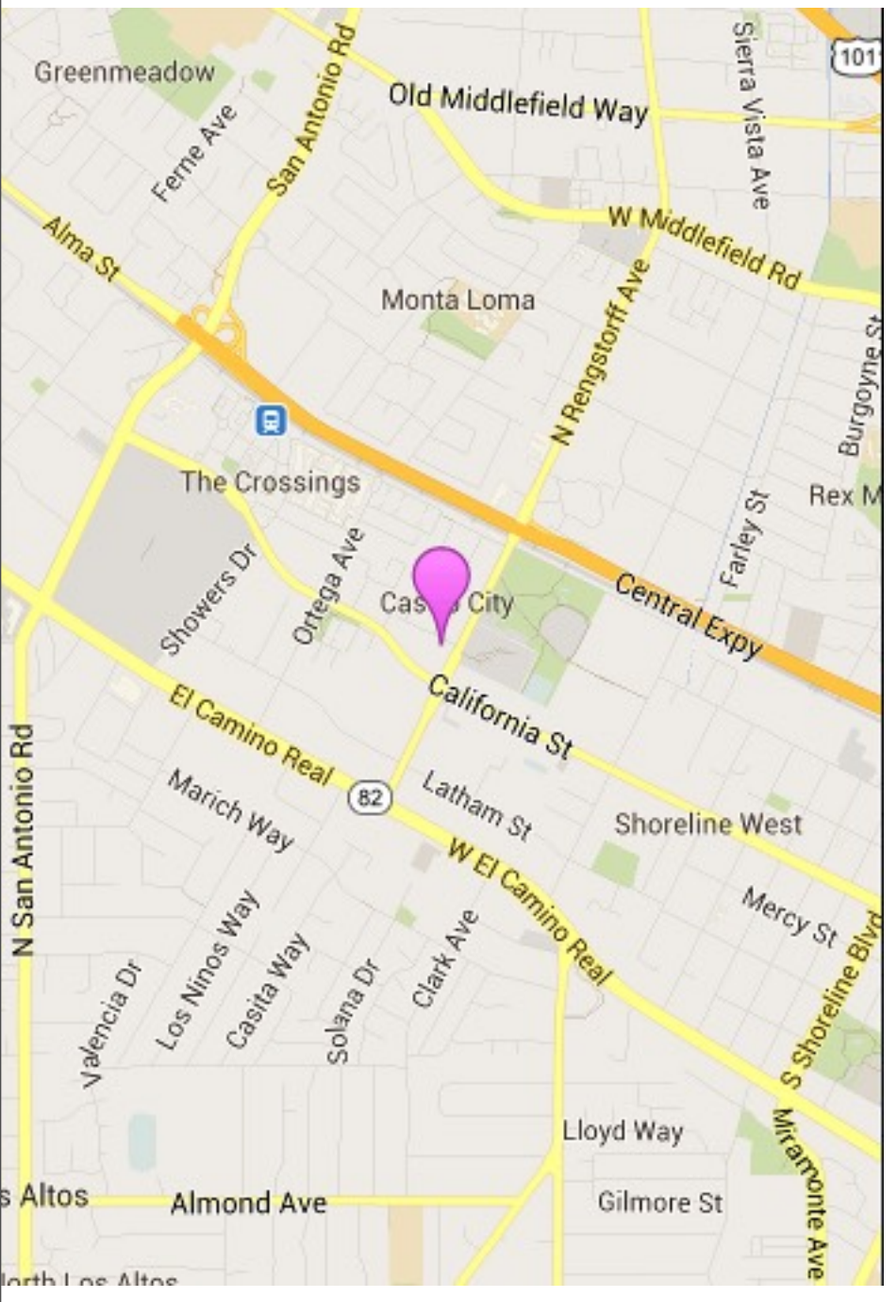
Zoom level 1 = Show entire world on screen

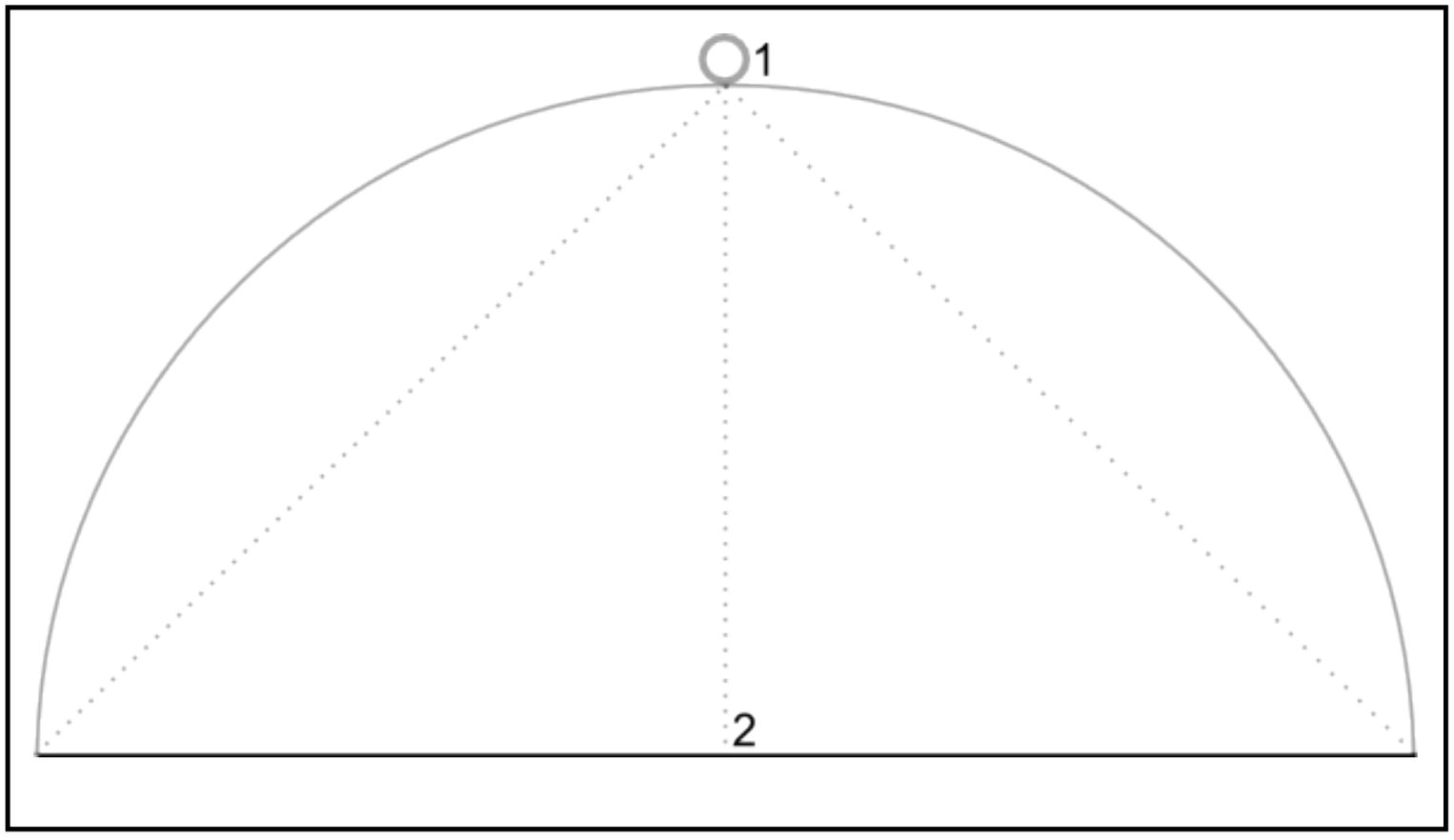
Increasing zoom level by 1

Doubles the width of the world on the screen

Tilt

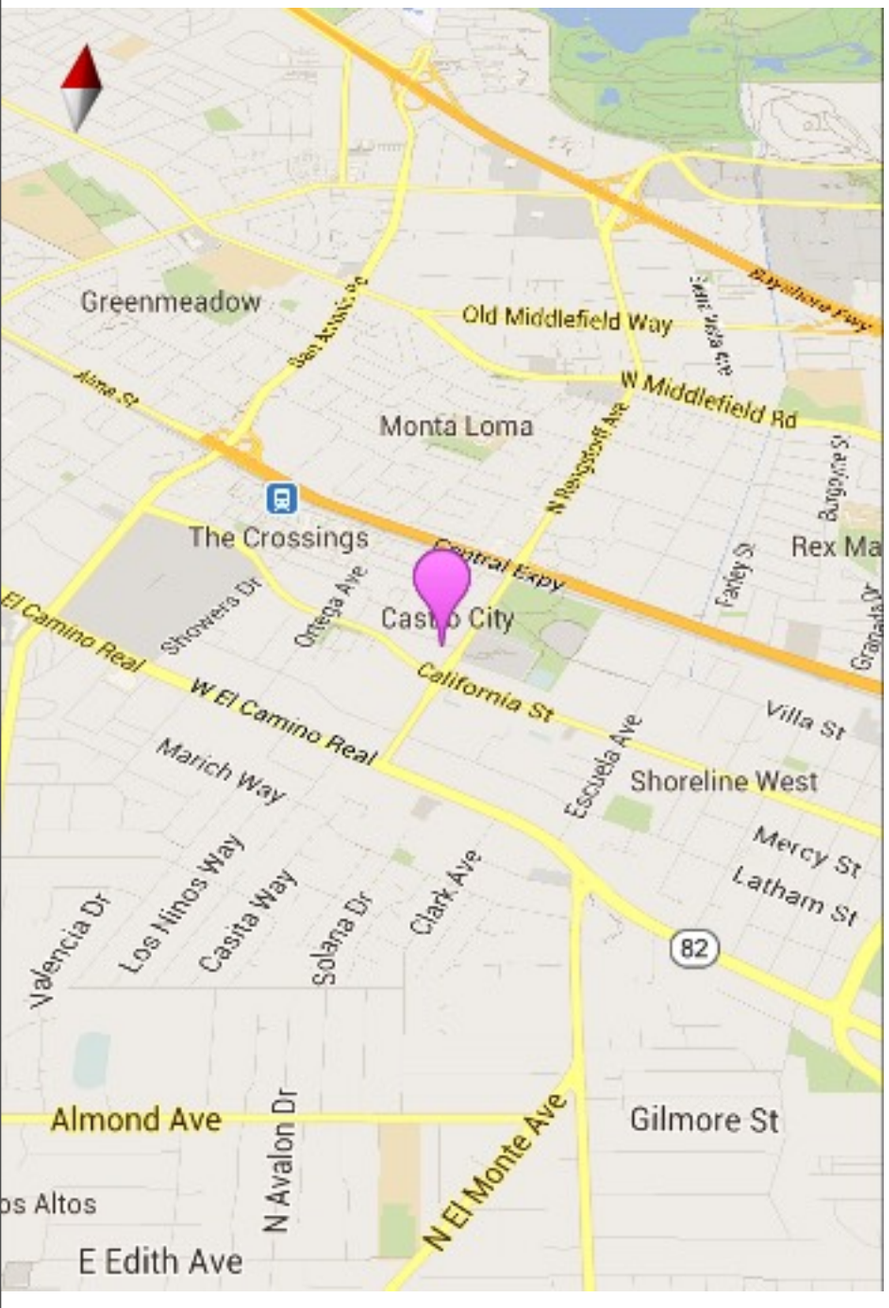
map:cameraTilt="0"

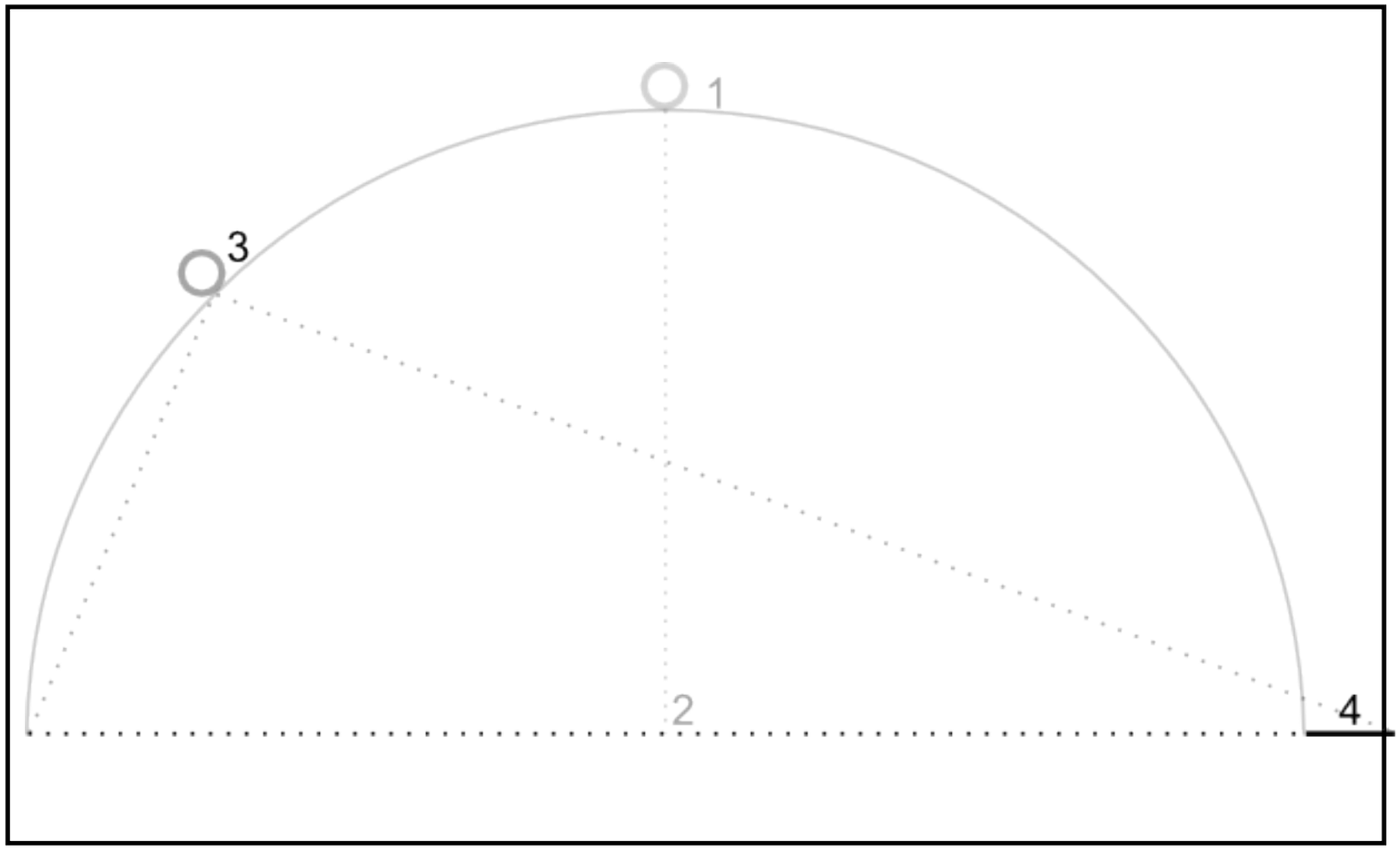




Tilt

map:cameraTilt="45"





Bearing (Orientation)

map:cameraBearing="112.5"

The direction in which a vertical line on the map points, measured in degrees clockwise from north

Markers


1:46
MapExample
Hardy Ave
1517
1517C
1517A
1537
1517F
1517E
1527
1515
1509
1505
1518A
1512A
1534
1532
1530
1504B
1504C
1502
1504
1504D
1502F
1504A
Entrance
Dr
4
3
2
1
KPBS,Copley
Telecommunications
Center
Lindo Paseo
Google

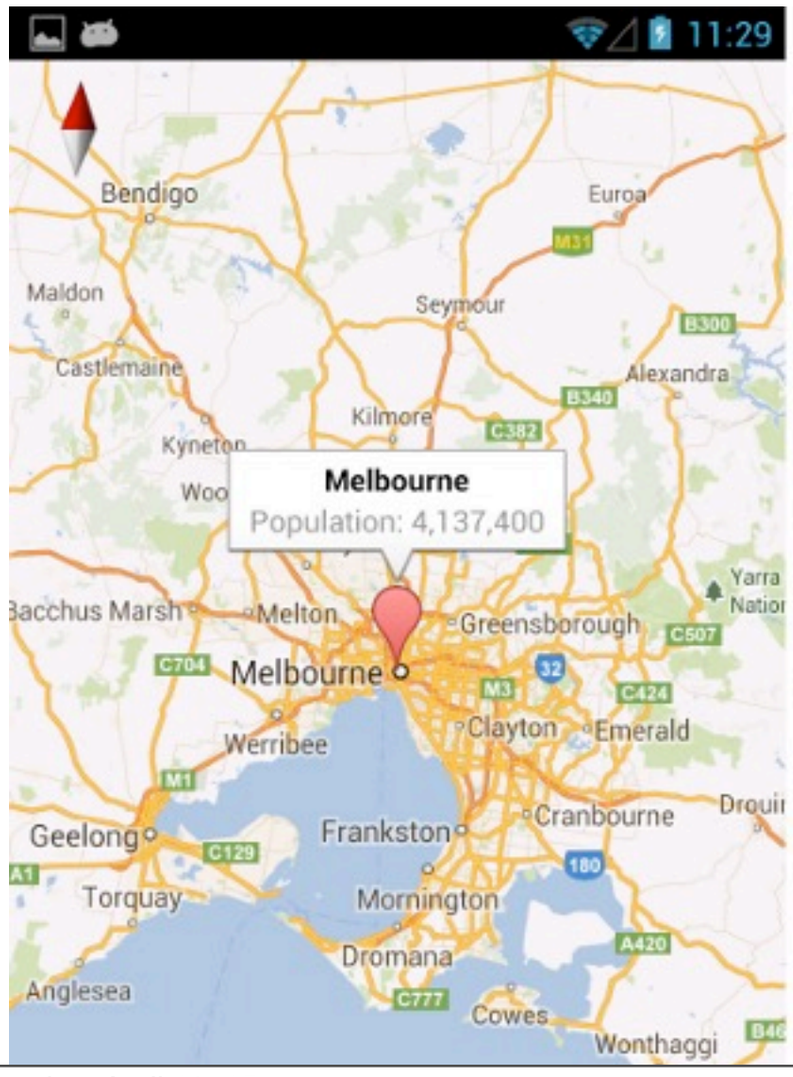

1:47
MapExample
Hardy Ave
Class is Here
1517
1517C
1517A
1537
1517F
1517E
1527
1515
1509
1505
1534
1532
1530
1504B
1504C
1502
1504
1504D
1502F
1504A
Entrance
Dr
4
3
2
1
KPBS,Copley
Telecommunications
Center
Lindo Paseo
Google

Setting the Marker

```
public class MainActivity extends Activity {

    static final LatLng GC1504 = new LatLng(32.772148, -117.072399);
    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

         mMap = ((MapFragment)
        getFragmentManager().findFragmentById(R.id.map)).getMap();
         MarkerOptions classRoom = new MarkerOptions()
                                                .position(GC1504)
                                                .title("Class is Near Here");
    mMap.addMarker(classRoom);
    }
}
```

Info Windows

```
static final LatLng MELBOURNE = new LatLng(-37.81319, 144.96298);
Marker melbourne = mMap.addMarker(new MarkerOptions()
                    .position(MELBOURNE)
                    .title("Melbourne")
                    .snippet("Population: 4,137,400"));
```

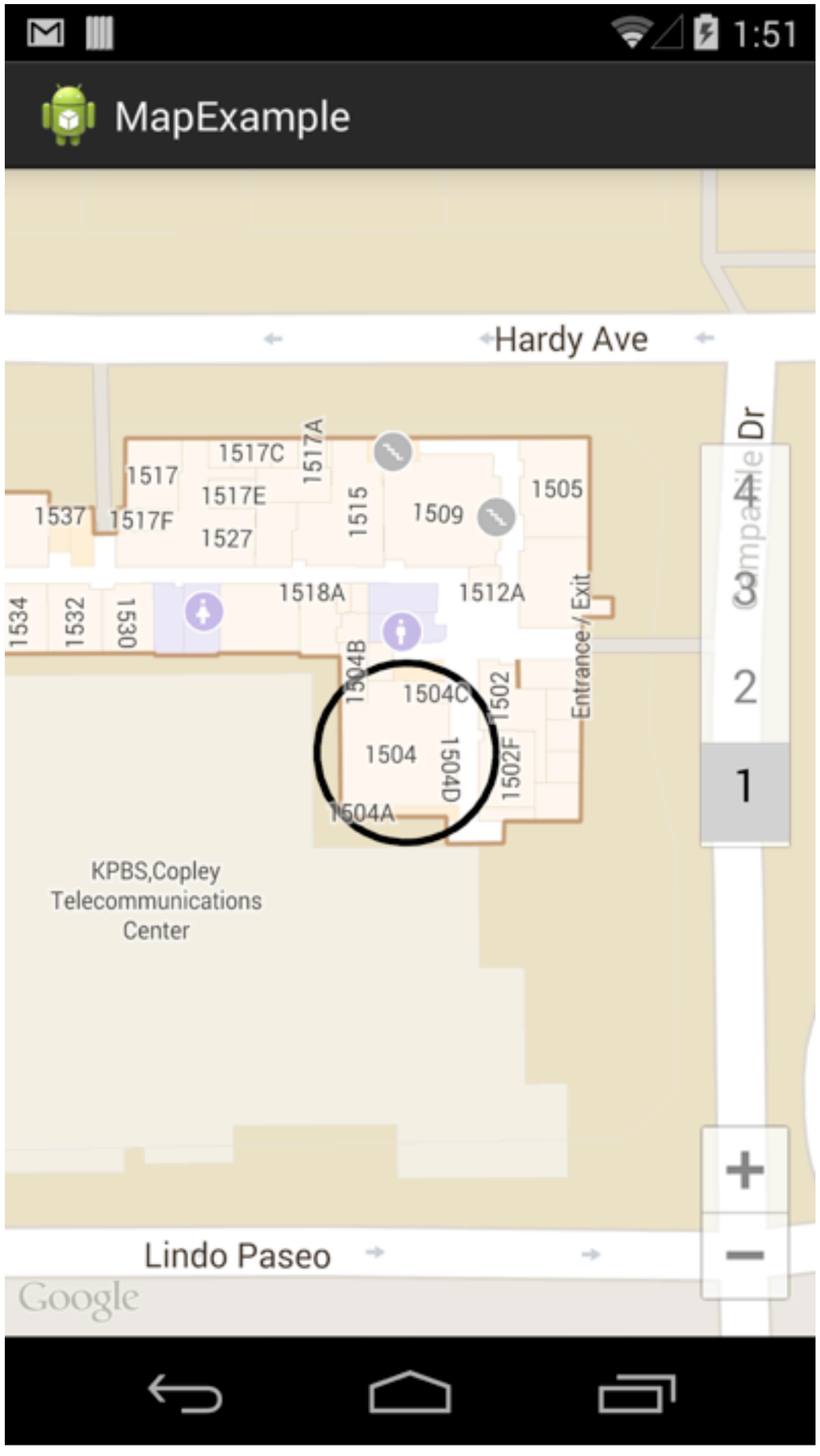


Shapes

Polylines

Polygons

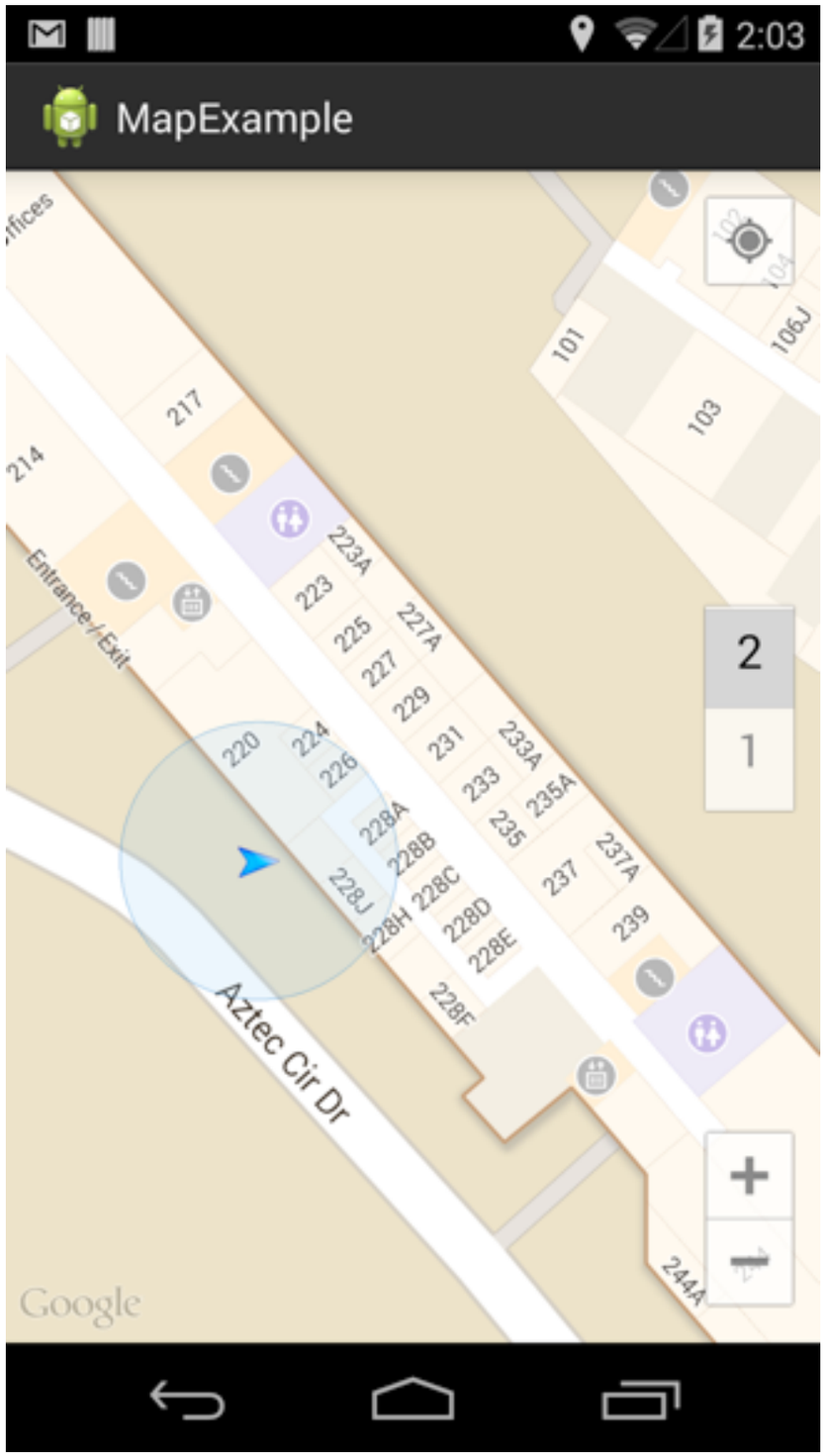
Circles

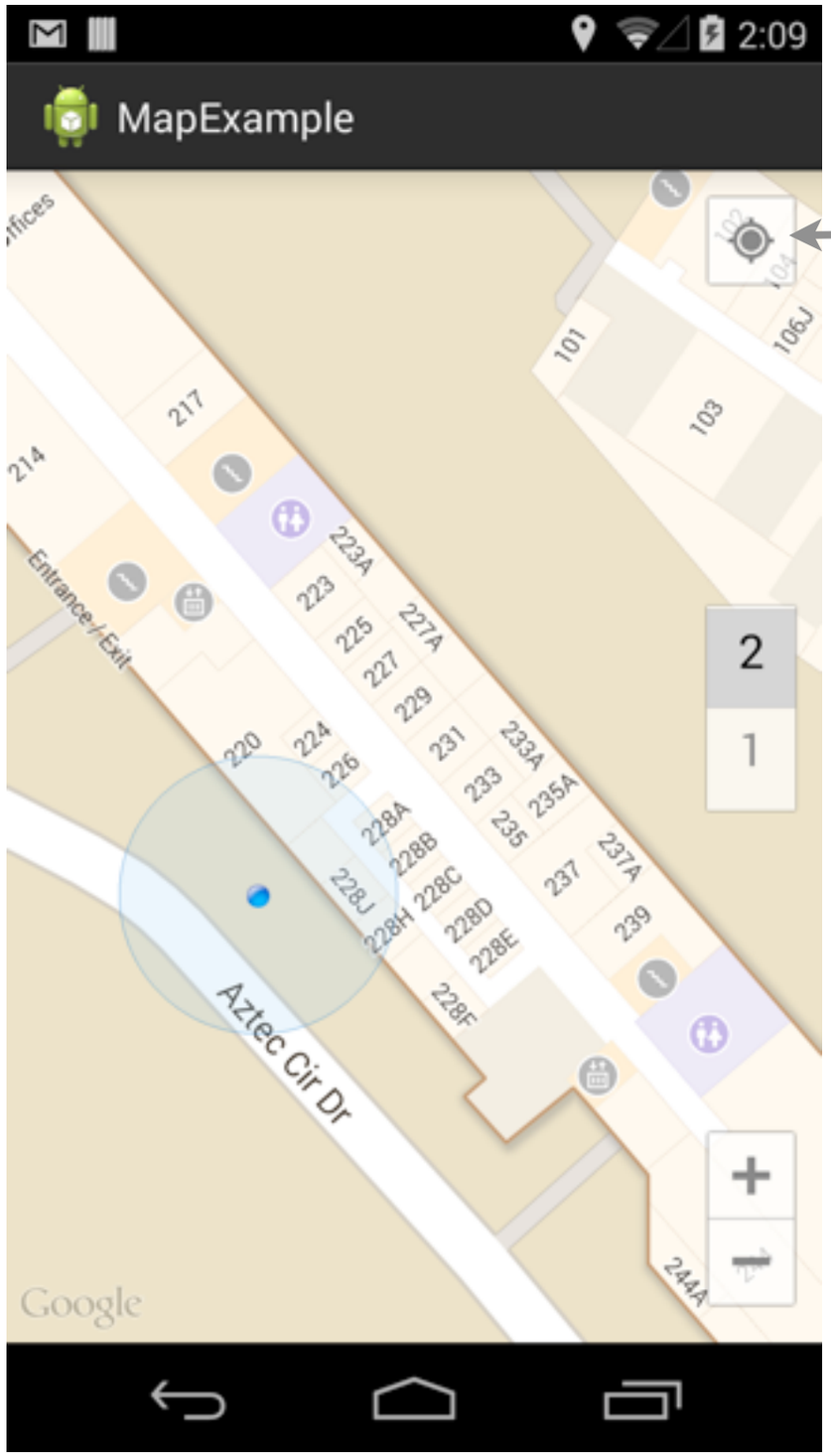


Circles

```
public class MainActivity extends Activity {

    static final LatLng GC1504 = new LatLng(32.772148, -117.072399);
    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CircleOptions circleOptions = new
CircleOptions().center(GC1504).radius(10);
                                                        // In meters

        Circle circle = mMap.addCircle(circleOptions);
    }
```

Device Location





Current Location

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mMap = ((MapFragment)
getFragmentManager().findFragmentById(R.id.map))
                    .getMap();
        mMap.setMyLocationEnabled(true);
        }
```