



# Android

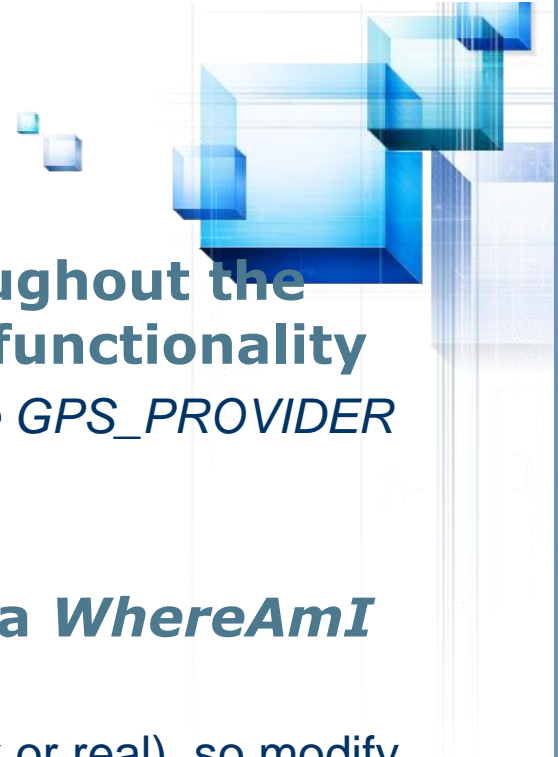


Location-Based Services

2010.04.29

***Database Laboratory  
Hanyang Univ.***

# ***Where Am I Example***



- ❖ **You will expand on this example throughout the chapter as you learn new geographic functionality**
  - *This example assumes that you have enabled the `GPS_PROVIDER` Location Provider*
  
- ❖ **Create a new *WhereAmI* project with a *WhereAmI* Activity**
  - This example uses the GPS provider (either mock or real), so modify the manifest file to include the uses-permission tag for `ACCESS_FINE_LOCATION`

# Where Am I Example

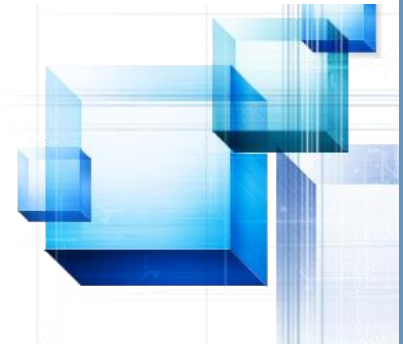


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.paad.whereami">
    <application android:icon="@drawable/icon">
        <activity android:name=".WhereAmI" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <uses-library android:name="com.google.android.maps"/>
    </application>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
</manifest>
```

- ❖ Modify the main.xml layout resource to include an *android:ID* attribute for the *TextView* control so that you can access it from within the Activity.

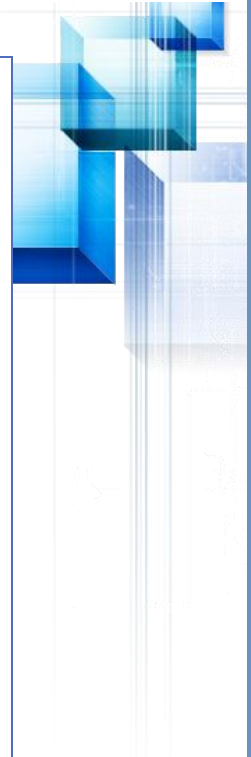
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/myLocationText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

# ***Where Am I Example***



- ❖ **Override the *onCreate* method of the *WhereAmI* Activity to get a reference to the Location Manager.**
  - Call *getLastKnownLocation* to get the last location fix value, and pass it in to the *updateWithNewLocation* method stub.

# Where Am I Example



```
package com.paad.whereami;

import java.util.List;

import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;

public class WhereAmI extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        LocationManager locationManager;
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        List<String> providers = locationManager.getProviders(true);

        for(String provider : providers) {
            locationManager.requestLocationUpdates(provider, 1000, 0, new LocationListener() {
                public void onLocationChanged(Location location) {}
                public void onProviderDisabled(String provider) {}
                public void onProviderEnabled(String provider) {}
                public void onStatusChanged(String provider, int status, Bundle extras) {}
            });

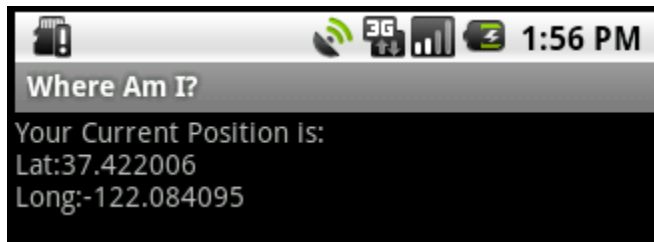
            Location location = locationManager.getLastKnownLocation(provider);
            updateWithNewLocation(location);
        }
    }
}
```

# Where Am I Example



- ❖ Fill in the *updateWithNewLocation* method to display the passed-in Location in the Text View by extracting the latitude and longitude values.

```
/** Update UI with a new location */  
private void updateWithNewLocation(Location location) {  
    TextView myLocationText = (TextView)findViewById(R.id.myLocationText);  
  
    String latLongString;  
  
    if (location != null) {  
        double lat = location.getLatitude();  
        double lng = location.getLongitude();  
        latLongString = "Lat:" + lat + "\nLong:" + lng;  
    } else {  
        latLongString = "No location found";  
    }  
  
    myLocationText.setText("Your Current Position is:\n" + latLongString);  
}
```



# ***Where Am I Example***



- ❖ **In the following example, “Where Am I?” is enhanced to track your current location by listening for location changes.**
  - Updates are restricted to one every 2 seconds, and only when movement of more than 10 meters has been detected

# Where Am I Example



- ❖ Update the *onCreate* method to find the best Location Provider that features high accuracy and draws as little power as possible.

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    LocationManager locationManager;  
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
  
    Criteria criteria = new Criteria();  
    criteria.setAccuracy(Criteria.ACCURACY_FINE);  
    criteria.setAltitudeRequired(false);  
    criteria.setBearingRequired(false);  
    criteria.setCostAllowed(true);  
    criteria.setPowerRequirement(Criteria.POWER_LOW);  
  
    String provider = locationManager.getBestProvider(criteria, true);  
  
    Location location = locationManager.getLastKnownLocation(provider);  
  
    updateWithNewLocation(location);  
}
```



# Where Am I Example



- ❖ Create a new *LocationListener* instance variable that fires the existing *updateWithNewLocation* method whenever a location change is detected

```
private final LocationListener locationListener = new LocationListener() {  
    public void onLocationChanged(Location location) {  
        updateWithNewLocation(location);  
    }  
  
    public void onProviderDisabled(String provider){  
        updateWithNewLocation(null);  
    }  
  
    public void onProviderEnabled(String provider){}  
  
    public void onStatusChanged(String provider, int status, Bundle extras) {}  
};
```

# Where Am I Example



## ❖ Return to *onCreate* and execute *requestLocationUpdates*, passing in the new *LocationListener* object

- It should listen for location changes every 2 seconds but fire only when it detects movement of more than 10 meters

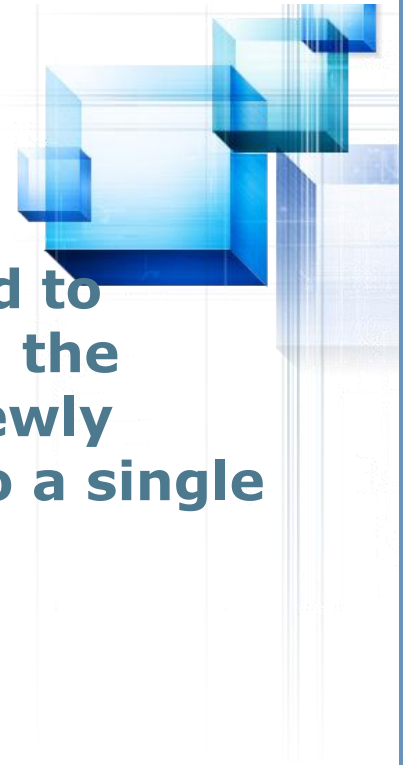
```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    LocationManager locationManager;  
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
  
    Criteria criteria = new Criteria();  
    criteria.setAccuracy(Criteria.ACCURACY_FINE);  
    criteria.setAltitudeRequired(false);  
    criteria.setBearingRequired(false);  
    criteria.setCostAllowed(true);  
    criteria.setPowerRequirement(Criteria.POWER_LOW);  
  
    String provider = locationManager.getBestProvider(criteria, true);  
  
    Location location = locationManager.getLastKnownLocation(provider);  
  
    updateWithNewLocation(location);  
  
    locationManager.requestLocationUpdates(provider, 2000, 10, locationListener);  
}
```

# ***Where Am I Example***



- ❖ **If you run the application and start changing the device location, you will see the Text View update accordingly**
- ❖ **Using the Geocoder, you can determine the street address at your current location**
- ❖ **you'll further extend the "Where Am I?" project to include and update the current street address whenever the device moves**

## ***Where Am I Example***



- ❖ **Modify the *updateWithNewLocation* method to instantiate a new Geocoder object, and call the *getFromLocation* method, passing in the newly received location and limiting the results to a single address.**

```
private void updateWithNewLocation(Location location) {
    // Update the map location.
    Double geoLat = location.getLatitude()*1E6;
    Double geoLng = location.getLongitude()*1E6;
    GeoPoint point = new GeoPoint(geoLat.intValue(), geoLng.intValue());

    mapController.animateTo(point);

    TextView myLocationText = (TextView)findViewById(R.id.myLocationText);
```

```
String latLongString;
String addressString = "No address found";
```

```
if (location != null) {
    double lat = location.getLatitude();
    double lng = location.getLongitude();
    latLongString = "Lat:" + lat + "\nLong:" + lng;

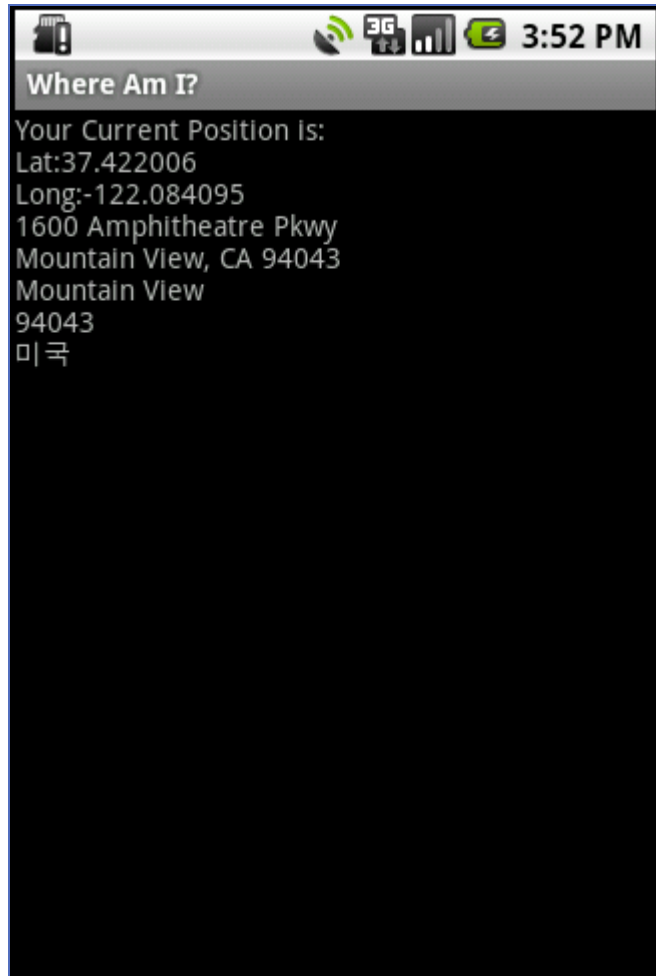
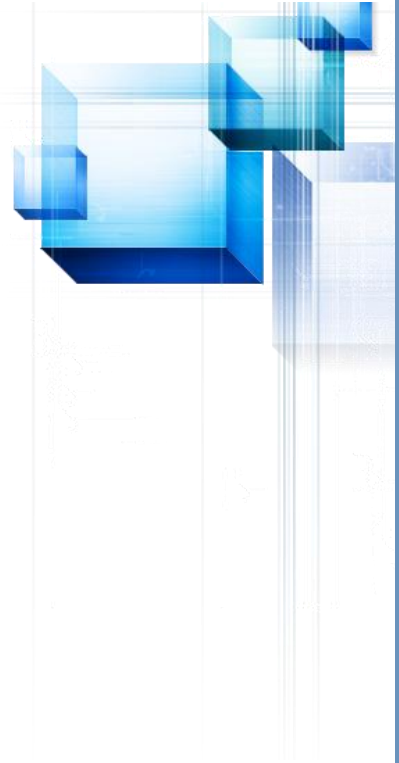
    Geocoder gc = new Geocoder(this, Locale.getDefault());
    try {
        List<Address> addresses = gc.getFromLocation(lat, lng, 1);
        StringBuilder sb = new StringBuilder();
        if (addresses.size() > 0) {
            Address address = addresses.get(0);

            for (int i = 0; i < address.getMaxAddressLineIndex(); i++)
                sb.append(address.getAddressLine(i)).append("\n");

            sb.append(address.getLocality()).append("\n");
            sb.append(address.getPostalCode()).append("\n");
            sb.append(address.getCountryName());
        }
        addressString = sb.toString();
    } catch (IOException e) {}
} else {
    latLongString = "No location found";
}
```

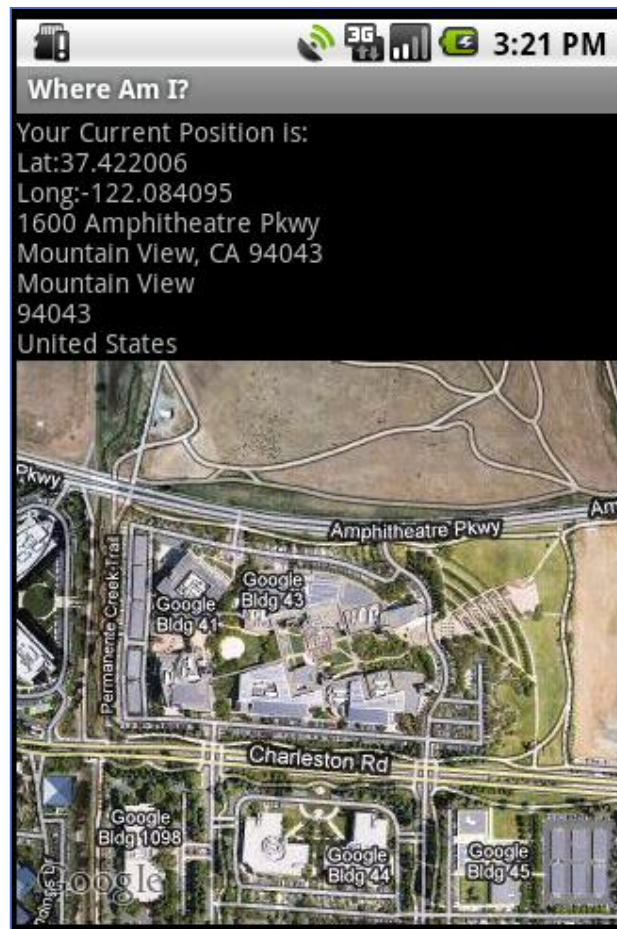
```
myLocationText.setText("Your Current Position is:\n" + latLongString + "\n" + addressString);
}
```

# ***Where Am I Example***



# ***Where Am I Example***

- ❖ **Now, Why not try Something!**
- ❖ **Let us use maps in our application**



## ***Obtain a Maps API Key***



- ❖ <http://code.google.com/intl/ko-KR/android/add-ons/google-apis/mapkey>.



# Hint



- ❖ **Note that you need to include a maps API key in order to use a Map View in your application**

```
<com.google.android.maps.MapView
    android:id="@+id/myMapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="true"
    android:apiKey="0iVAgvPMH-8sODnD1fIMLLA3jnVd1cYtJS1IXWQ"
/>
```

- ❖ **Before you can use the latitude and longitude values stored in the Location objects used by the locationbased services, you'll need to convert them to microdegrees and store them as GeoPoints, as shown in the following code snippet:**

```
Double geoLat = location.getLatitude()*1E6;
Double geoLng = location.getLongitude()*1E6;
GeoPoint point = new GeoPoint(geoLat.intValue(), geoLng.intValue());
```