

Programming Mobile Applications with Android

22-26 September, Albacete, Spain

Jesus Martínez-Gómez



Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Introduction to advanced android capabilities
 - Maps and locations.- How to use them and limitations.
 - Sensors.- Using sensors to obtain real-time data
 - External Data.- How to access files, databases and content providers
 - Android Lab V

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - In this lesson, we will learn:
 - What advanced capabilities are included in Android and how to use them
 - How to use localization and maps in our applications
 - How to access data not stored in our applications

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Introduction to advanced android capabilities
 - We have seen how to develop Android applications similar to desktop ones
 - Multimedia elements
 - Graphical user interfaces
 - Life Cycle
 - Access to the camera
- ... but the execution in a mobile phone allows for new and more powerful functionalities

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Introduction to advanced android capabilities
 - Mobile phones are equipped with
 - Cameras
 - GPS
 - Tactile screen
 - Temperature and humidity sensors
 - Accelerometer and compass
 - Access to Google services
 - Networking
 - Mobile capabilities: contacts, calls, sms, etc

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Introduction to advanced android capabilities
 - Using some of these elements, we can:
 - Develop an application that generates a map with the images acquired in a specific environment
 - Create new gestures to speed-up repetitive actions, like the introduction of contacts in the agenda
 - Create an application for notifying when the temperature arises a threshold
 - Develop applications to acquired the quality of our domestic wireless network and then present some statistics
 - ...

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.- How to use them and limitations.
 - Localization services are basic for the development of useful applications
 - Where can I find the nearest fuel station?
 - Where can I find the place where I took a photograph
 - Load advertisements only when the localization is the appropriate
 - For instance, a Restaurant 2x1 ticket when the user is close to the closest Restaurant

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.
 - How to retrieve the current location?
 - We will develop both a service and an activity
 - Service to provide localization capabilities
 - Activity to require access to the service
 - Permissions

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

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

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

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

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


Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.
 - How to retrieve the current location? - Service Class
 - Implements LocationListener
 - A set of methods should be implemented *onLocationChanged(Location location)*
 - Key variables
 - Location myLocation
 - Context myContext
 - getLocation() method returns the location by using the locationManager obtained from myContext

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.- How to use them and limitations.
 - How to retrieve the current location? - Service Class

```
locationManager = (LocationManager) myContext.getSystemService(LOCATION_SERVICE);  
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,6000,10, this);  
myLocation = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
myLatitude = myLocation.getLatitude();  
myLongitude= myLocation.getLongitude();
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.- How to use them and limitations.
 - How to retrieve the current location? - Activity
 - Includes an object of the service class that provides the localization
 - We can now obtain out latitude and longitude when needed
 - If the GPS is not enable, it will fail
 - There should be checked that GPS is working or open the localization settings

Programming Mobile Applications with Android

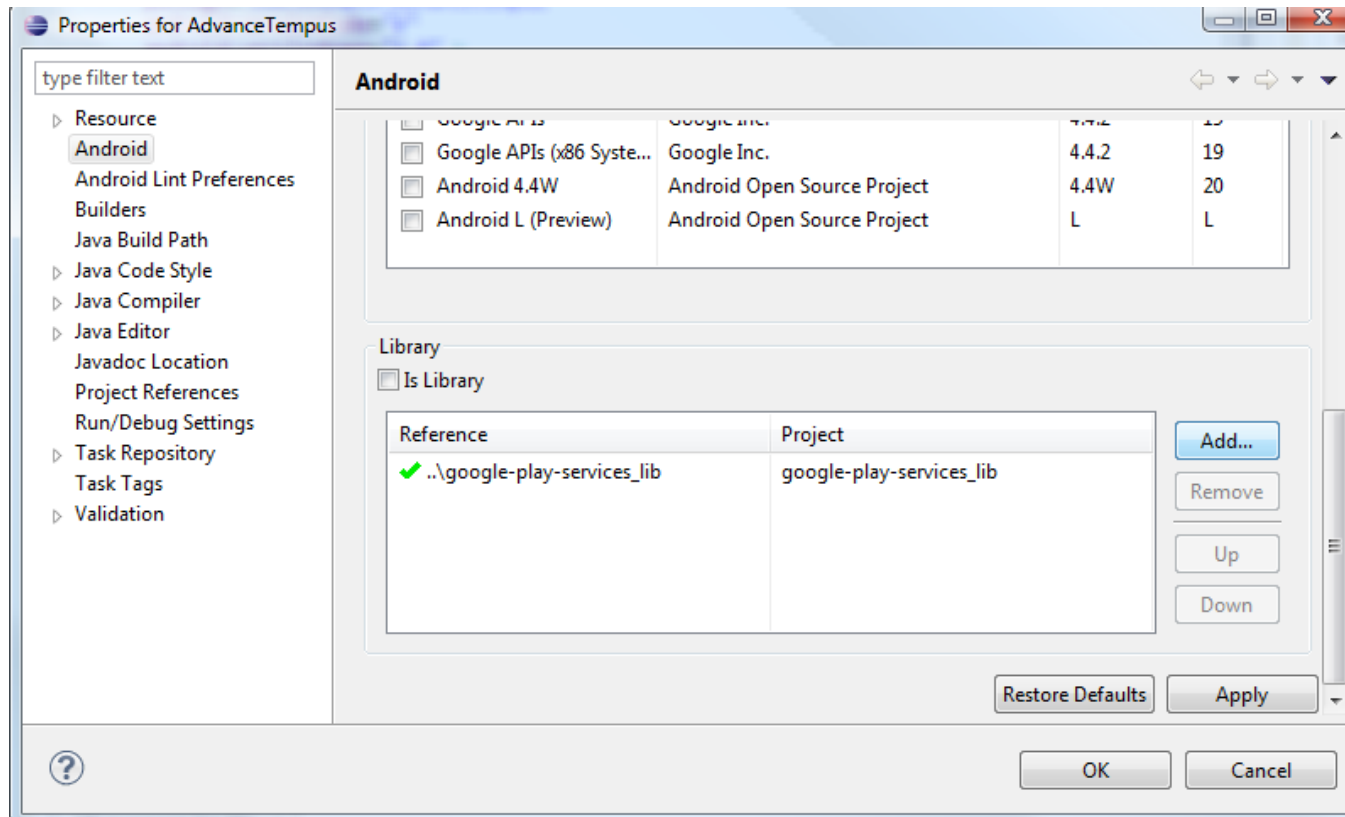
- Lesson 5.- Android Advanced
 - Maps and locations.- How to use them and limitations.
 - Once we can access to our location, the following step consist in using maps
 - We can add maps based on Google Maps data
 - The Google Maps API provides access to:
 - Maps Servers
 - Data Download
 - Map Display
 - Touch Gestures

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.
 - We will need to install the Google Play services SDK
 - <https://developer.android.com/google/play-services/setup.html>
 - First, copy the folder in our workSpace
 - android-sdks\extras\google\google_play_services\libproject
 - Then, add this project as library
 - Right button → properties → Android

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations.



Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- Maps and locations.

- We need an Android Certificate and the Google Maps API key

- <https://code.google.com/apis/console/?noredirect#project:1017833389>
 - Services → activate the Google Maps Android API v2

 Google Compute Engine Resource Views API 	<input type="checkbox"/> OFF	Courtesy limit: 1,000,000 requests/day
 Google Contacts CardDAV API 	<input type="checkbox"/> OFF	Courtesy limit: 20,000,000 requests/day
 Google Maps Android API v2 	<input checked="" type="checkbox"/> ON	
 Google Maps Coordinate API 	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
 Google Maps Embed API 	<input type="checkbox"/> OFF	Courtesy limit: 2,000,000 requests/day
 Google Maps Engine API 	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- Maps and locations.

- We need an Android Certificate and the Google Maps API key

- We need to use our SHA1 certificate fingerprint

- Windows -> Preferences -> Android -> Build

- And then use it to obtain the Key and modify the manifest file and include the following lines

```
<meta-data android:name="com.google.android.maps.v2.API_KEY"
android:value="AlzaSyC9Y40rcnX....." />
```

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


Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations
 - We now should have all these permissions

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

```
<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_NETWORK_STATE"/>
```

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

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

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

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

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

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations
 - Now we can add and manage google maps in our applications
 - Useful webpage → <http://www.gps-coordinates.net/>
 - We can obtain the coordinates of any worldwide location
 - Layout

```
<fragment  
android:id="@+id/map"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
class="com.google.android.gms.maps.MapFragment" />
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- Maps and locations

- Activity code

- *GoogleMap map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap()*

- The object GoogleMap can now be modified to move the camera, make zoom, add markers, etc

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Maps and locations
 - Adding markers

```
final LatLng ESII = new LatLng(38.97878612469428, -1.8558686971664429);  
map.addMarker(new MarkerOptions().position(ESII).title("University"));
```

- Modifying the map visualization

```
map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);  
map.setMapType(GoogleMap.MAP_TYPE_NORMAL);  
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);  
map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- Maps and locations

- Move the camera to an specific position

```
final LatLng l3A = new LatLng(38.979019654695456, -1.854119896888733);  
map.moveCamera(CameraUpdateFactory.newLatLng(l3A));
```

- Disable zoom and other UI settings

```
map.getUiSettings().setCompassEnabled(true);  
map.getUiSettings().setZoomControlsEnabled(false);
```

- Animate the camera movements

```
map.animateCamera(CameraUpdateFactory.zoomTo(15), 2000, null);
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - In addition to the camera or GPS, we can access to some sensors like the accelerometer, proximity, etc
 - If we want to use a sensor, we should firstly detect if we can access without problems

```
SensorManager mSensorManager;
```

```
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

```
if(mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null
```

```
Toast.makeText(this,"Sucess", Toast.LENGTH_SHORT).show();
```

```
else
```

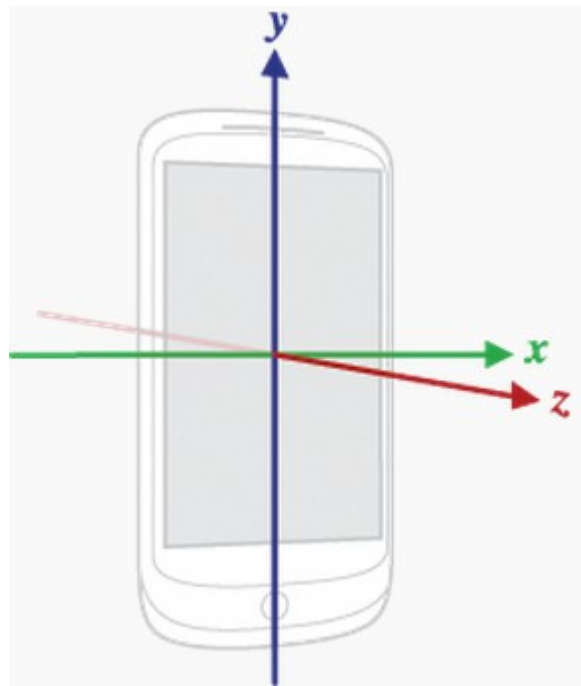
```
Toast.makeText(this,"Fail", Toast.LENGTH_SHORT).show();
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Sensors are usually managed to monitor some events
 - For instance: check that the temperature is below certain threshold
 - We need to implement 2 callback methods by making our activity implement the class `SensorsEventListener`
 - `onAccuracyChanged()`
 - `onSensorChanged()`
 - We need also register a `SensorEventListener` for an specific sensor
 - `registerListener(SensorEventListener l, Sensor s, int rate);`

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
- Sensor Coordinate System



Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Some basics
 - Register just the sensors that are needed
 - Unregister the listeners as soon as possible
 - Test your code on real devices, not the emulator
 - Verify sensors before using them

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Example A.- Accessing the accelerometer
 - Detect shake movements
 - Detect steps
 - Draw figures on the air
 - ...

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Example A.- Accessing the accelerometer
 - Global Variables

```
private SensorManager mSensorManager;  
private Sensor mAccelerometer;
```

- Initialization

```
mSensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
mAccelerometer =  
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
mSensorManager.registerListener(this, mAccelerometer,  
SensorManager.SENSOR_DELAY_NORMAL);
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Example A.- Accessing the accelerometer
 - Capture the changes

```
public void onSensorChanged(SensorEvent event)
{
float x = event.values[0]; float y = event.values[1]; float z = event.values[2];
}
```

- Unregistering on pause

```
protected void onPause()
{
super.onPause();
mSensorManager.unregisterListener(this);
}
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Example B.- Accessing the proximity sensor
 - Global Variables

```
private SensorManager mSensorManager;  
private Sensor mProximitySensor;
```

- Initialization

```
mSensorManager = (SensorManager)  
getService(Context.SENSOR_SERVICE);  
mProximitySensor =  
mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);  
mSensorManager.registerListener(this, mProximitySensor,  
SensorManager.SENSOR_DELAY_NORMAL);
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Sensors.- Using sensors to obtain real-time data
 - Example B.- Accessing the proximity sensor
 - Capture the changes

```
public void onSensorChanged(SensorEvent event)  
{  
    Toast.makeText(this, "Distance "+String.valueOf(event.values[0]),  
    Toast.LENGTH_SHORT).show();  
}
```

- Unregistering on pause

```
protected void onPause()  
{  
    super.onPause();  
    mSensorManager.unregisterListener(this);    }
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.- How to access files, databases and content providers
 - Not all the data included in our applications can be stored when released
 - Size requirements
 - Dynamic changes
 - We can find several alternatives

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.- Alternatives
 - Shared Preferences
 - Store private primitive data in key-value pairs.
 - Internal Storage
 - Store private data on the device memory.
 - External Storage
 - Store public data on the shared external storage.
 - SQLite Databases
 - Store structured data in a private database.
 - Network Connection
 - Store data on the web with your own network server.

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- External Data.

- Shared Preferences

- This is the most basic type of storage and allows to save any type of primitive data

- Save

```
SharedPreferences settings = getSharedPreferences("TempusPref", 0);  
SharedPreferences.Editor editor = settings.edit();  
editor.putBoolean("myBoolVar", true);  
editor.commit();
```

- Load

```
SharedPreferences settings = getSharedPreferences("TempusPref", 0);  
boolean varLoaded = settings.getBoolean("myBoolVar", false);  
setBoolVar(varLoaded);
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - Internal Storage
 - Similar to the file access in other programming environments
 - Files saved with our applications cannot be accessed by the rest of the applications (by default)

```
String FILENAME = "fileToSave";
```

```
String string = "text to save";
```

```
FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
```

```
fos.write(string.getBytes());
```

```
fos.close();
```

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - External Storage
 - Files that need to be written in external SD cards or similar, instead of internal storage
 - We need additional permissions to access external storage

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - Databases
 - Similar to the web pages, all the dynamic content of an android application can be obtained from a database
 - The technology used for most Android applications is SQLite, with considerable differences with respect to traditional databases languages
 - Network Connections
 - We can use the network connections to obtain data from the Internet, as we did to obtain images

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - Content Providers
 - Access to structured sets of data
 - Permission management
 - Key elements
 - ContentResolver → For the access to content providers
 - URI → Identification of content providers
 - Basis steps to retrieve data from a content provider
 - Request the read access permission for the provider.
 - Define the code that sends a query to the provider.

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced

- External Data.

- Content Providers

- Request the read access permission for the provider.

- Using the manifest file

- <uses-permission android:name="android.permission.READ_USER_DICTIONARY">*

- Define the code that sends a query to the provider.

- ContentResolver.Query()

- Insert data

- Update data

- Delete data

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - Content Providers
 - Data types: integer, long integer (long), floating point, long floating point (double)
 - Building a content provider is a complex task and needs to be performed only when necessary
 - Design data storage
 - Design content URIs
 - Implement the ContentProvider Class
 - Implement the ContentProvider MIME types
 - ...
 - Permissions and Access

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - External Data.
 - Content Providers.- Contact providers
 - Android component for the management of our contacts
 - Three types of data about a person
 - Basic steps to retrieve a list of contacts
 - Permissions
 - `<uses-permission android:name="android.permission.READ_CONTACTS" />`
 - Define layout elements for visualization
 - Follow the guidelines from <http://developer.android.com/training/contacts-provider/retrieve-names.html>

Programming Mobile Applications with Android

- Lesson 5.- Android Advanced
 - Android Lab V.- Create, compile and execute an advanced application with localization, external storage and sensors processing.
 - Follow the instructions to create an application with access to the Google Maps Api with sensor processing and external storage of data

Programming Mobile Applications with Android

22-26 September, Albacete, Spain

Jesus Martínez-Gómez

