Native Android Apps

INF5750/9750 - Lecture 7 (Part I)

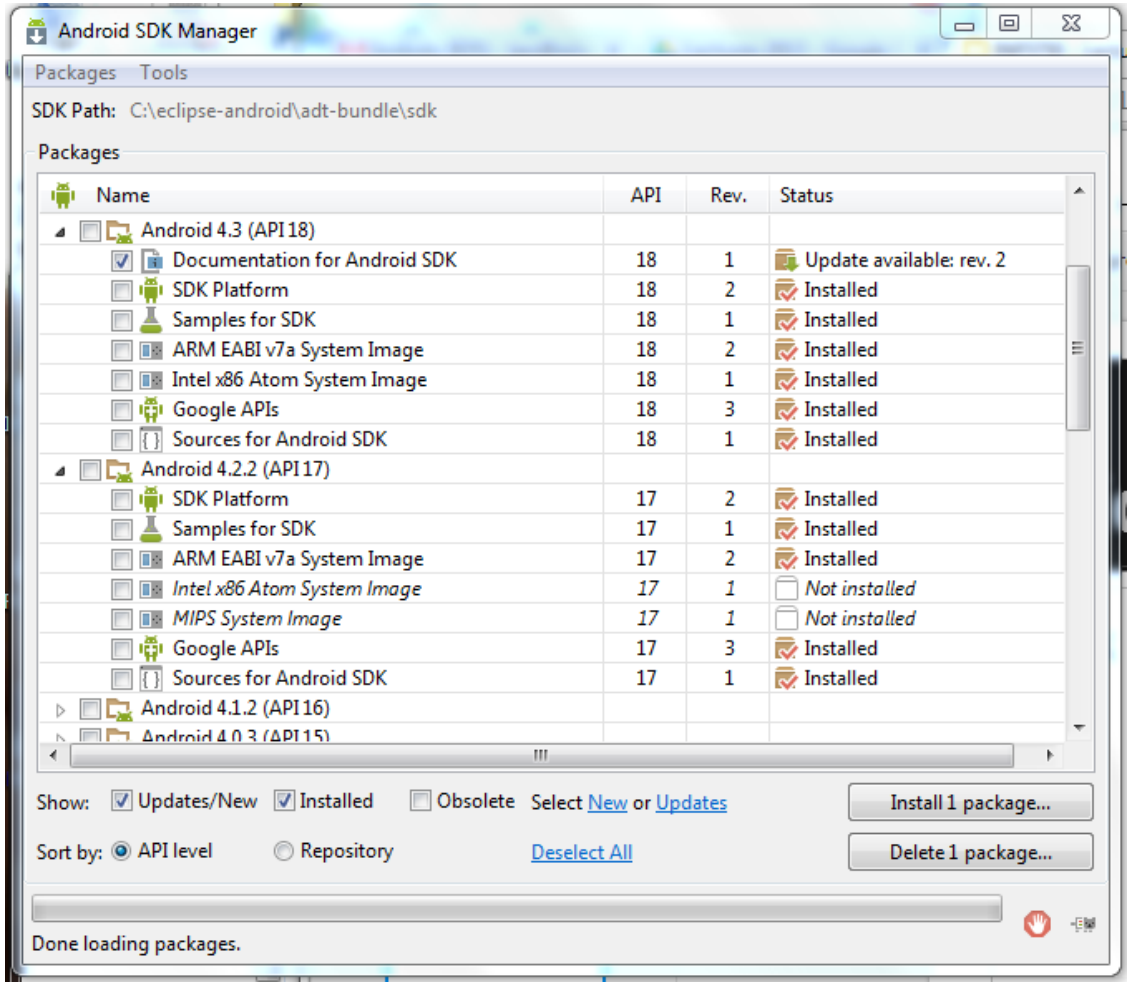
Lecture contents

- Setting up the development environment
- Android apps basics
- Examples

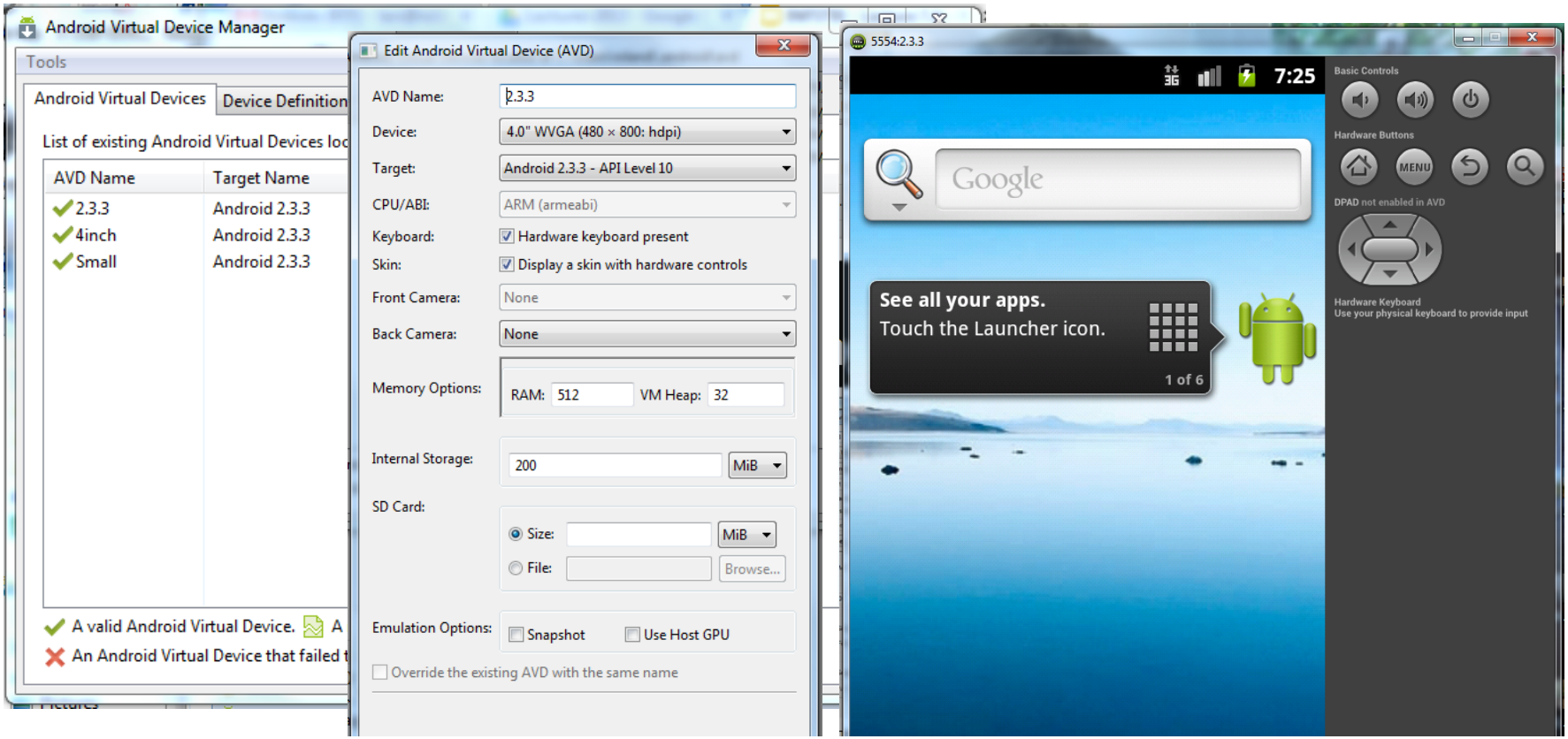
Development environment

- Android SDK - ADT (Android Dev Tools)
  - Contains APIs and libraries. Multiple versions
  - Build and packaging tools
  - Emulator (runs the app virtually on your computer)
  - ADB (Android Debug Bridge) - used to communicate with emulator and real phones (over USB).
- Windows, Mac OS X, Linux
- Installed in separate directory, but can come pre-bundled with IDE (or plug in afterwards)
- http://developer.android.com/sdk/index.html

SDK Manager


Android SDK Manager
Packages Tools
SDK Path: C:\eclipse-android\adt-bundle\sdk
Packages
Name API Rev. Status
Android 4.3 (API 18)
Documentation for Android SDK 18 1 Update available: rev. 2
SDK Platform 18 2 Installed
Samples for SDK 18 1 Installed
ARM EABI v7a System Image 18 2 Installed
Intel x86 Atom System Image 18 1 Installed
Google APIs 18 3 Installed
Sources for Android SDK 18 1 Installed
Android 4.2.2 (API 17)
SDK Platform 17 2 Installed
Samples for SDK 17 1 Installed
ARM EABI v7a System Image 17 2 Installed
Intel x86 Atom System Image 17 1 Not installed
MIPS System Image 17 1 Not installed
Google APIs 17 3 Installed
Sources for Android SDK 17 1 Installed
Android 4.1.2 (API 16)
Show: Updates/New Installed Obsolete Select New or Updates
Install 1 package...
Sort by: API level Repository Deselect All
Delete 1 package...
Done loading packages.

AVD Manager



You can select ARM or Intel Atom as the CPU. Intel may be faster on Intel-platforms, but requires installation of some additional libraries to get optimized speed.
http://software.intel.com/en-us/articles/speeding-up-the-android-emulator-on-intel-architecture

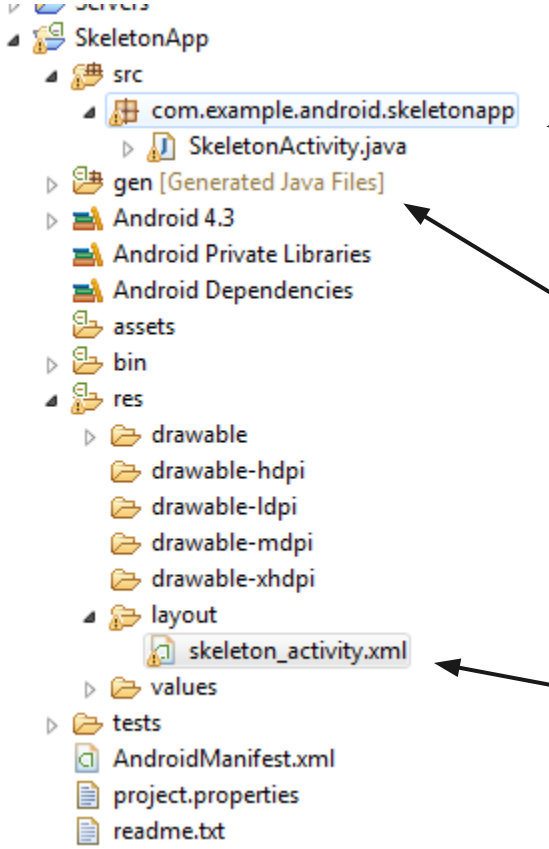
IDE Integration

- Some IDEs come bundled with Android Integr
- For stock Eclipse, need to install ADT Plugin
- developer.android.com/sdk/installing/installing-adt.html
- With the ADT plugin, you can :
  - Import Android projects
  - Design the user interface layout inside Eclipse
  - Edit Android resource files
  - Build and package Android apps inside Eclipse
  - Run emulator from Eclipse

Android basics

- User interface is created using Views. The view layout is defined in a xml file.
- User interface logic is implemented in a Java class extending Activity. Each View typically has its own Activity.
- Intent is a central concept for passing info.
- Background jobs, network tasks etc should not be done in the UI thread. There are ways to spawn off background-jobs and then update the UI afterwards.

File structure

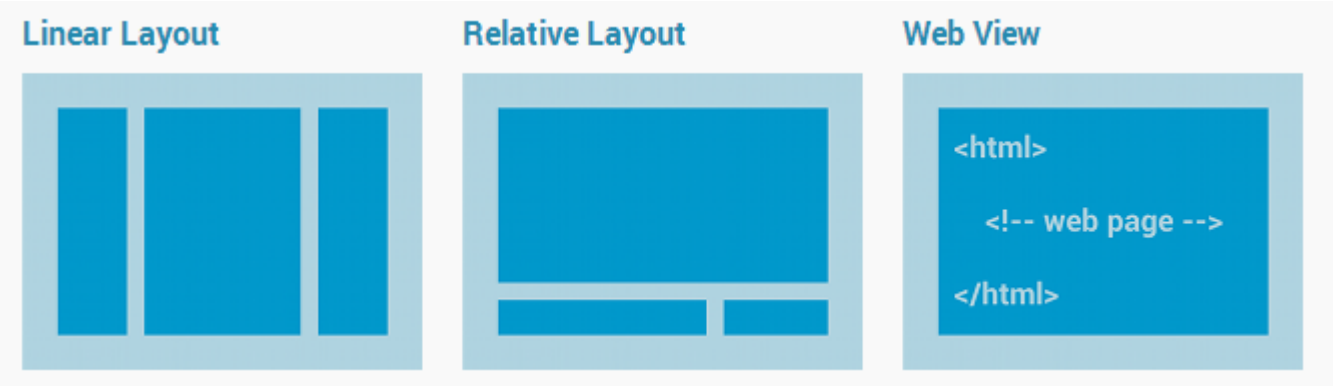


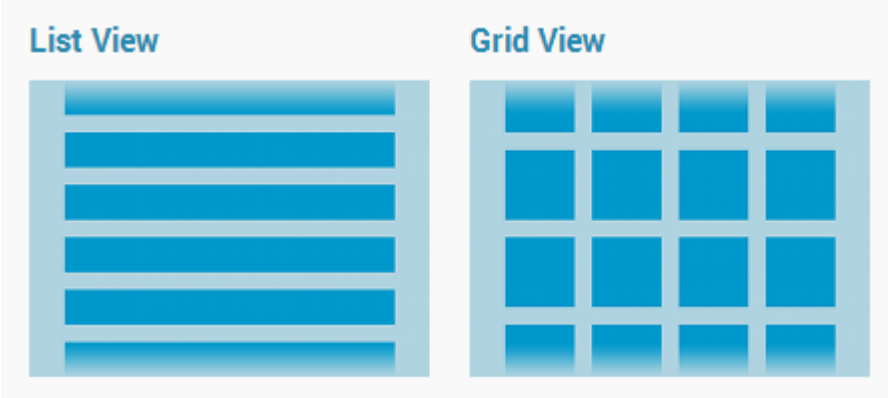
Activities and other Java classes.

Automatically generated Java classes

XML-based layout files and other resources

Common layouts

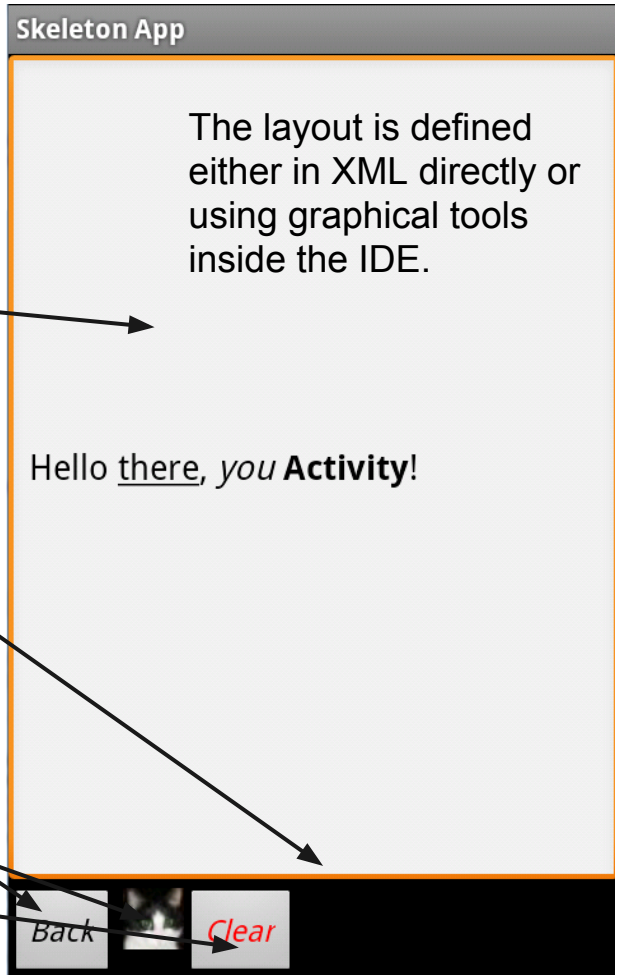




UI components are defined inside layouts. Use an XML file to define how the layout.

View - layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="
match_parent"
    android:orientation="vertical">
    <EditText android:id="@+id/editor"
        android:layout_width="match_parent" android:layout_height="0dip"...
    </EditText>
    <LinearLayout  android:layout_width="wrap_content" ..>
        <Button android:id="@+id/back" style="@style/ActionButton"
            android:text="@string/back" />
        <ImageView
            android:id="@+id/image"
            ... />
        <Button android:id="@+id/clear" style="@style/ActionButton"
            android:text="@string/clear" android:textColor="@color/red" />
    </LinearLayout>
</LinearLayout>
```

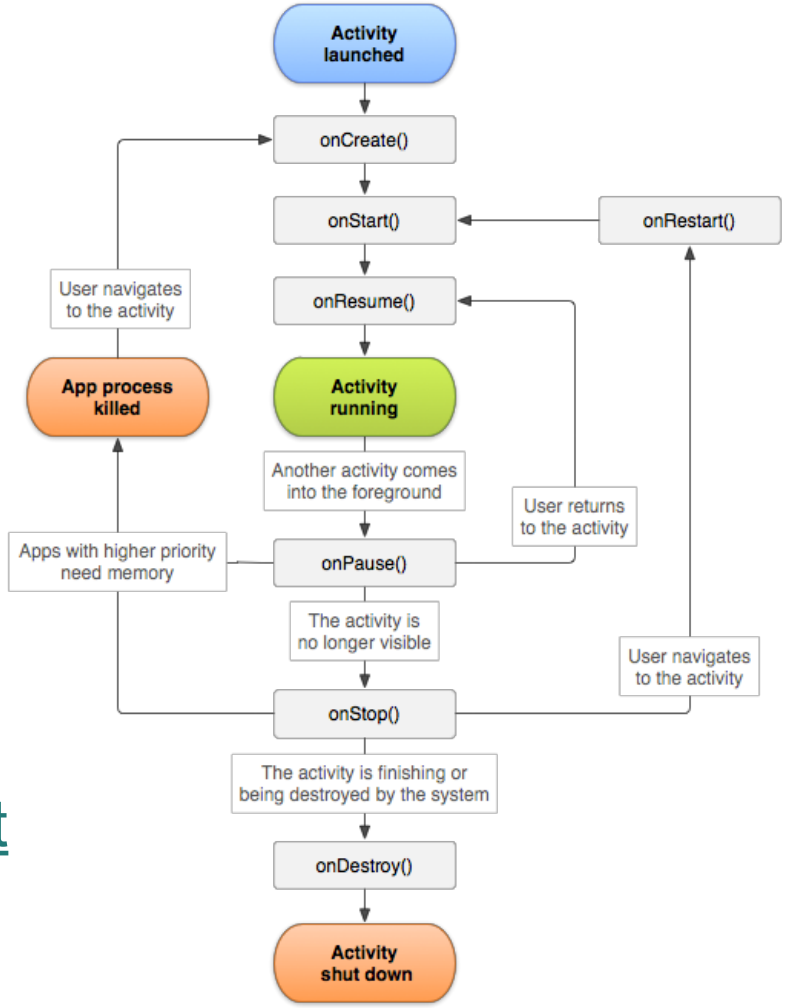


Android Activity methods

onCreate(Bundle) is where you initialize your activity. Call setContentView(int) with a layout resource defining your UI, and use findViewById(int) to retrieve widgets.

onPause() gets called when the user leaves the activity.

http://developer.android.com/reference/android/app/Activity.html



Activity

```
public class SkeletonActivity extends Activity {
    private EditText mEditor; // Define variables for your GUI elements
…
/** Called with the activity is first created. */
public void onCreate(Bundle savedInstanceState) {
        setContentView(R.layout.skeleton_activity); // Tell Activity which view to use
        mEditor = (EditText) findViewById(R.id.editor); // Initialize GUI elements
        ((Button) findViewById(R.id.back)).setOnClickListener(mBackListener);
        ((Button) findViewById(R.id.clear)).setOnClickListener(mClearListener);
        mEditor.setText(getText(R.string.main_label));
 }
…
OnClickListener mBackListener = new OnClickListener() { // Implement the button listener
        public void onClick(View v) {
            finish();
        }
```

Intents

- Android is built up by Activities, Services and Broadcast receivers
- The messages between these components are called ‘Intents’.
- Example intents: Incoming call, Send an SMS, Change network state, Battery information, Screen turned off etc.
- You can also create own Intents
- Use the AndroidManifest.xml to listen to Intents

Threading

- Network access not in UI thread.Use for example AsyncTask to tell Android to run network access outside UI thread.
- Only the UI thread can update the UI. Instead, for example call .runOnUiThread:

```
YourActivityName.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                Toast.makeText(YourActivityName.this, "Toast!!!", Toast.LENGTH_SHORT).show();
            }
        });
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:windowSoftInputMode="adjustPan"
    package="org.apache.cordova.example" android:versionName="1.0" android:versionCode="1" android:hardwareAccelerated="true">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
     <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <application android:icon="@drawable/icon" android:label="@string/app_name"
    android:hardwareAccelerated="true"
    android:debuggable="true">
        <activity android:name="example" android:label="@string/app_name"
                android:theme="@android:style/Theme.Black.NoTitleBar"
                android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
    </activity>
  </application>
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="17"/>
</manifest>
```